

IDENTIFYING HUMAN FINGERPRINTS BY USING OPTICAL COHERENCE
TOMOGRAPHY IMAGE WITH DEEP LEARNING



PAPAWIT NONGHKUNSAN

A Thesis Submitted in Partial Fulfilment of the Requirement for the
Degree of Master of Science in Applied Physics
Suranaree University of Technology
Academic Year 2022

การระบุอัตลักษณ์ของบุคคลจากภาพถ่ายลายนิ้วมือออปติคัลโคฮีเรนซ์
โทโมกราฟฟีด้วยการเรียนรู้เชิงลึก



นายปภาวิชญ์ หนองขุนสาร

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาฟิสิกส์ประยุกต์
มหาวิทยาลัยเทคโนโลยีสุรนารี
ปีการศึกษา 2565

IDENTIFYING HUMAN FINGERPRINTS BY USING OPTICAL COHERENCE
TOMOGRAPHY IMAGE WITH DEEP LEARNING

Suranaree University of Technology has approved this thesis submitted in partial fulfillment of the requirements for a Master's Degree.

Thesis Examining Committee



(Dr. Wiwat Nuansing)

Chairperson



(Dr. Ittipon Fongkaew)

Member (Thesis Advisor)



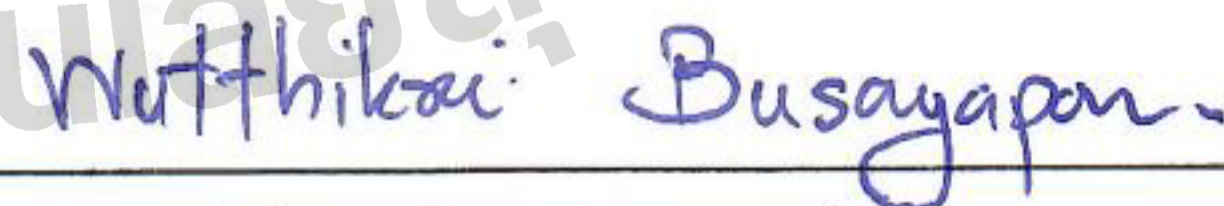
(Assoc. Dr. Panomsak Meemon)

Member (Thesis Co-advisor)



(Assoc. Dr. Sirapat Boonkrong)

Member



(Dr. Wutthikrai Busayaporn)

Member



(Prof. Dr. Santi Maensiri)

Dean of Institute of Science



(Assoc. Prof. Dr. Chatchai Jothityangkoon)

Vice Rector for Academic Affairs
and Quality Assurance

ปภาวิชญ์ หนองขุนสาร : การระบุอัตลักษณ์ของบุคคลจากภาพถ่ายลายนิ้วมือออฟติคัลโคฮีเรนซ์
โทโมกราฟี ด้วยการเรียนรู้เชิงลึก (IDENTIFYING HUMAN FINGERPRINTS BY USING
OPTICAL COHERENCE TOMOGRAPHY IMAGE WITH DEEP LEARNING)
อาจารย์ที่ปรึกษา : อาจารย์ ดร.อิทธิพล ฟองแก้ว, 103 หน้า.

คำสำคัญ: ออฟติคัลโคฮีเรนซ์โทโมกราฟี (โอซีที), การเรียนรู้เชิงลึก, โครงข่ายประสาทแบบคอนโวลูชัน

สแกนเนอร์ลายนิ้วมือแบบ 2D ดั้งเดิมมีความเสี่ยงต่อการปลอมแปลงลายนิ้วมือ เนื่องจากระบบ
จะพิจารณาข้อมูลเฉพาะพื้นผิว เพื่อแก้สถานการณ์นี้ งานนี้เสนอวิธีการระบุตัวตนด้วยลายนิ้วมือ โดยใช้
ภาพแบบตัดขวางจากเครื่อง Optical Coherence Tomography (OCT) ภาพเหล่านี้ถูกประมวลผลโดย
ใช้ Convolutional Neural Network (CNN) ซึ่งเริ่มต้น ด้วยการเทรนข้อมูลด้วยโครงสร้างโมเดลพื้นฐาน
แล้วจึงปรับปรุงโมเดลด้วยการเรียนรู้การถ่ายโอน โดยใช้ชุดคุณสมบัติที่กว้างขวางของโมเดลที่มีการฝึกฝน
ผ่านโครงสร้างที่มีความซับซ้อนไว้แล้ว ซึ่งรวมถึง InceptionV3, ResNet50, VGG16, และ Xception
เพื่อเพิ่มประสิทธิภาพการทำนาย จึงใช้หลักการของโมเดลเอนเซมเบิลโดยใช้วิธีการลงคะแนนส่วนใหญ่
สำหรับผลลัพธ์สุดท้าย วิธีการนี้ลดความเสี่ยงต่อการปลอมแปลงลายนิ้วมือลงอย่างมาก และให้ความ
แม่นยำที่สูงกว่า โดยทำความสำเร็จ 100% ในการทดสอบโดยใช้ภาพถ่ายลายนิ้วมือจาก 12 ผู้ใช้ งานวิจัย
นี้ได้เน้นถึงศักยภาพของภาพ OCT และการเรียนรู้เชิงลึกในการเสนอเพื่อประยุกต์ใช้ในความปลอดภัย
และความเชื่อถือได้ของระบบการยืนยันตัวตนด้วยลายนิ้วมือ

โดยสรุปการประยุกต์ใช้ OCT การเรียนรู้เชิงลึกและโมเดลเอนเซมเบิลเปิดทางสู่มาตรการเพิ่ม
ความปลอดภัยอีกทางเลือก ด้วยความแม่นยำ 100% ที่ได้รับการทดสอบ สามารถแสดงถึงการก้าว
หน้าที่มีประสิทธิภาพกว่าเทคนิคการสแกนเนอร์ลายนิ้วมือแบบ 2D ดั้งเดิม ซึ่งย้ำถึงศักยภาพที่มากขึ้น
ของวิธีการในงานนี้ ในการเพิ่มประสิทธิภาพและความน่าเชื่อถือของระบบการยืนยันตัวตนด้วยลายนิ้วมือ

สาขาวิชาฟิสิกส์

ปีการศึกษา 2565

ลายมือชื่อนักศึกษา ปภาวิชญ์ หนองขุนสาร

ลายมือชื่ออาจารย์ที่ปรึกษา อิทธิพล

ลายมือชื่ออาจารย์ที่ปรึกษาร่วม สมชาย

PAPAWIT NONGKHUNSAN : IDENTIFYING HUMAN FINGERPRINTS BY USING OPTICAL COHERENCE TOMOGRAPHY IMAGE WITH DEEP LEARNING

THESIS ADVISOR : ITTIPON FONGKAEW, Ph.D. 103 PP.

Keyword: Optical coherence tomography (OCT), Deep learning (DL), Convolutional neural network (CNN)

Traditional 2D fingerprint scanners are vulnerable to spoofing due to their reliance on surface data. In response, this work proposes a novel fingerprint identification technique using cross-sectional images from Optical Coherence Tomography (OCT). These images are processed using a Convolutional Neural Network (CNN), initially trained on a conventional architecture. The model is then enhanced with transfer learning, utilizing the broad feature sets of pre-trained models including InceptionV3, ResNet50, VGG16, and Xception. To further boost prediction accuracy, this research apply an ensemble model principle using a majority voting method for the final output. This approach greatly reduces susceptibility to spoofing and offers superior accuracy, achieving 100% in 12 identity fingerprints. this research highlights the potential of OCT images and deep learning proposed for the applied with the security and reliability of fingerprint-based identity verification systems.

In conclusion, innovative application of OCT, deep learning, and ensemble models pave the way for optional cybersecurity measures. By achieving 100% accuracy in a tests, this work demonstrate a tangible advancement over traditional 2D fingerprint scanning techniques, underlining the significant potential approach in bolstering the effectiveness and reliability of fingerprint-based identity verification systems.

School of Physics

Academic Year 2022

Student's Signature Papa wit Nongkhunsan

Advisor's Signature Ittipon Fongkaew

Co-advisor's Signature Ittipon Fongkaew

ACKNOWLEDGEMENTS

I am indebted to my project advisor, Dr. Ittipon Fongkaew, for his unwavering support and direction regarding the programming aspects of this research. His expertise and dedication were crucial to the successful completion of this project.

Dr. Panomsak Meemon and his team at the School of Physics, Institute of Science, Suranaree University of Technology have made invaluable contributions to the OCT hardware and image data management. Their expertise and assistance significantly enriched this investigation.

In addition, I would like to thank Mr. Jesada Saetiew and Ms. Jiraporn Saenjae for their assistance in preparing the OCT equipment and their support throughout the data collection process.

I would also like to thank my colleagues and fellow researchers at Suranaree University of Technology for their collaboration and thought-provoking discussions, which contributed to my development as a scholar and enriched my research experience.

Papawit Nongkhunsan

มหาวิทยาลัยเทคโนโลยีสุรนารี

CONTENTS

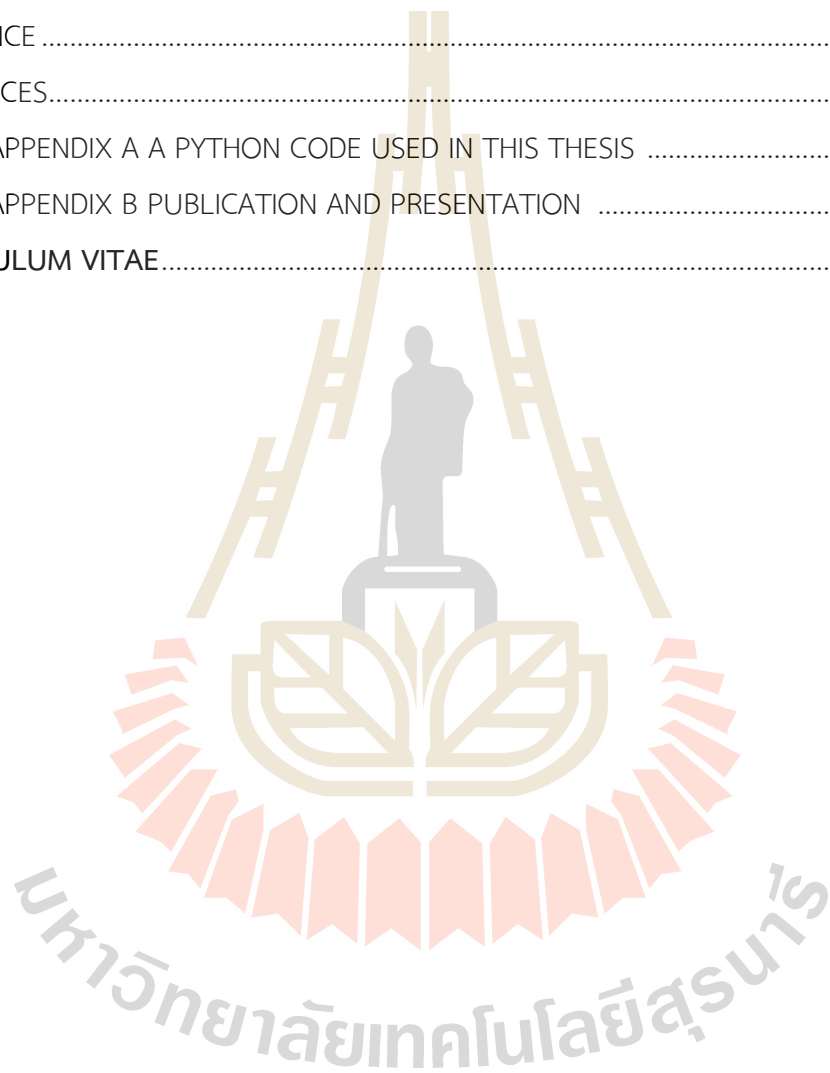
| | Page |
|--|----------|
| ABSTRACT IN THAI..... | I |
| ABSTRACT IN ENGLISH..... | II |
| ACKNOWLEDGEMENTS..... | III |
| CONTENTS..... | IV |
| LIST OF TABLES..... | VII |
| LIST OF FIGURES..... | IX |
| CHAPTER | |
| I Introduction | 1 |
| 1.1 Background and motivation..... | 1 |
| 1.2 Objectives of the research | 2 |
| 1.3 Scope and Limitations of the Study..... | 3 |
| 1.4 Expected results..... | 4 |
| II Literature review | 5 |
| 2.1 Fingerprint scanner | 5 |
| 2.1.1 capacitive sensor..... | 6 |
| 2.1.2 Optical scanner..... | 6 |
| 2.1.3 Ultrasonic sensor..... | 7 |
| 2.2 Optical coherence tomography..... | 8 |
| 2.2.1 Michelson interferometry | 8 |
| 2.2.2 Signal processing for image generated..... | 9 |
| 2.2.3 Data augmentation..... | 9 |
| 2.2.4 Presentation attack..... | 12 |
| 2.3 Deep learning..... | 12 |
| 2.3.1 Convolutional neural networks..... | 14 |

CONTENTS (Continued)

| | Page |
|--|-------------|
| 2.3.2 Artificial neural networks | 18 |
| 2.3.2 Transfer learning | 20 |
| 2.3.3 Ensemble model | 25 |
| 2.3.4 Performance evaluation | 28 |
| 2.4 Explainable artificial intelligence | 31 |
| 2.4.1 SHapley Additive exPlanations (SHAP) | 31 |
| 2.4.2 LIME (Local Interpretable Model-agnostic Explanations) | 32 |
| 2.4.3 Grad-CAM (Gradient-weighted Class Activation Mapping) | 32 |
| III Methodology | 34 |
| 3.1 Preparing data | 34 |
| 3.2 Model Architecture and Selection | 36 |
| 3.3 Training and validation procedures | 37 |
| 3.4 Model Evaluation and Comparison | 39 |
| 3.5 Experimental Setup and hardware specifications | 39 |
| IV Result | 41 |
| 4.1 Performance Evaluation Model | 41 |
| 4.2 Model comparison | 51 |
| 4.3 Analysis of Receiver Operating Characteristic (ROC) Curves | 53 |
| 4.4 Explainable model | 60 |
| 4.4 Image size effect | 62 |
| 4.5 Application | 63 |
| 4.6 Discussion of Limitations and Future Directions | 64 |

CONTENTS (Continued)

| | Page |
|--|------|
| V Conclusion | 66 |
| REFERENCE | 68 |
| APPENDICES..... | 74 |
| APPENDIX A A PYTHON CODE USED IN THIS THESIS | 75 |
| APPENDIX B PUBLICATION AND PRESENTATION | 86 |
| CURRICULUM VITAE..... | 104 |



LIST OF TABLES

| Table | Page |
|--|------|
| 1 Confusion matrix..... | 28 |
| 2 The parameter settings for four TL networks and a custom network..... | 38 |
| 3 Table of confusion matrix table for ResNet50 model. | 47 |
| 4 Table of confusion matrix table for Xception model..... | 48 |
| 5 Table of confusion matrix table for VGG16 model. | 49 |
| 6 Table of confusion matrix table for InceptionV3 model..... | 50 |
| 7 Table of confusion matrix table for custom model..... | 51 |
| 8 Comparison of Classification report between five models and voting method..... | 52 |
| 9 Explainable model table shows the feature selection by SHAP..... | 61 |
| 10 Table of classification reports for different image sizes. | 62 |

LIST OF FIGURES

| Figure | Page |
|--|------|
| 1 Schematic of working principle of capacitive scanner. | 6 |
| 2 Schematic of working principle of an optical fingerprint scanner. | 6 |
| 3 Schematic of working principle of an ultrasonicl fingerprint scanner..... | 7 |
| 4 Schematic of Michelson interferometer..... | 9 |
| 5 Image processing which A)raw data B) contrast adjust with noise filter (median filter) C) and D) Heat mapping. | 10 |
| 6 B-scan image with the dept information of fingertip. | 12 |
| 7 Deep learning: combining of feature extraction and prediction..... | 13 |
| 8 Convolution calculation. | 16 |
| 9 Convolution operation (a) zero padding (b) padding is 1..... | 17 |
| 10 Max pooling..... | 18 |
| 11 Basic artificial nueron (Zilouchian, A. and Jamshidi, M., 2001). | 19 |
| 12 diagram of transfer learning..... | 21 |
| 13 Resnet-50 model architecture (Ridha Ilyas, B., Beladgham, M., Merit, K. and taleb-ahmed, A., 2019). | 22 |
| 14 Xception neural network architecture diagrams of the model (Chollet, F., 2017)..... | 23 |
| 15 VGG16 neural network architecture diagrams of the model (Loukadakis, M., Cano, J. and O'Boyle, M., 2018)..... | 24 |
| 16 InceptionV3 neural network architecture diagrams of the model (Szegedy, Christian, Vanhoucke, Vincent, Ioffe, Sergey, Shlens, Jon and Wojna, Zbigniew, 2016)..... | 25 |

LIST OF FIGURES (Continued)

| Figure | Page |
|--|------|
| 17 Bagging technique schematic | 26 |
| 18 Voting technique schematic | 27 |
| 19 Boosting technique schematic | 27 |
| 20 AUC-ROC curve | 30 |
| 21 Example of finger OCT captured | 35 |
| 22 In-house optical coherence tomography | 35 |
| 23 Neural network architecture diagrams of a model | 37 |
| 24 Learning curve of RestNet50 pre-trained model | 42 |
| 25 Learning curve of Xception pre-trained model | 43 |
| 26 Learning curve of VGG16 pre-trained model | 44 |
| 27 Learning curve of InceptionV3 pre-trained model | 45 |
| 28 Learning curve of custom model | 46 |
| 29 ROC and AUC can visualize the performance of ResNet50 | 54 |
| 30 ROC and AUC cuve visualize the performance of Xception model | 55 |
| 31 ROC and AUC cuve visualize the performance of VGG16 model | 56 |
| 32 ROC and AUC cuve visualize the performance of InceptionV3 | 57 |
| 33 ROC and AUC curve visualize the performance of the custom model | 58 |
| 34 ROC and AUC cuve visualize the performance of Ensemble model | 59 |
| 35 Example for the application workflow diagram | 64 |

CHAPTER I

INTRODUCTION

1.1 Background and motivation

Nowadays, with the system of authentication required in every industry, the primary concern with personal data is privacy; bioinformatics has been utilized as a sort of protection in this regard. The traditional bioinformatics for fingerprint scanner techniques is capacitive technology, optical technology, and ultrasonic technology (Memon, S., Sepasian, M. and Balachandran, W., 2009). These traditional methods are widely used in every sector, for instance, mobile phones, accessing the security system, banking sector, and even identifying person at border control. The ridges on the finger used for personal identification are scanned in only two dimensions by these technologies. Additionally, the data was primarily saved as a template using traditional methods. When a user tries to access the system, the system compares the templates and scores the match using a threshold. Other systems will be dismissed as imposters if a score exceeds the cutoff, allowing the user to use the system as authentic, accordingly, a traditional scanner with access based on a matching template showed that it still has vulnerabilities (Biggio, B., Akhtar, Z., Fumera, G., Marcialis, G. L. and Roli, F., 2012). To address the issue, research suggests three-dimensional technology such as Optical coherence tomography (OCT) has efficient against the non-authentic fingerprint attack by detecting the spoofs finger and bonafide finger with a true detection rate (TRD) of 99.73% (Chugh, T. and Jain, A. K., 2019a, 2019b). So, classification by internal fingertip images could prevent hacking systems by classifying the real and fake fingerprint images obtained from OCT with very high accuracy.

OCT is a non-invasive imaging technique that uses light waves to capture detailed images of internal structures (Huang, D., Swanson, E. A., Lin, C. P., Schuman, J. S., Stinson, W. G., Chang, W.... 1991). In the field of biometrics, OCT has been used to capture images of the internal structures of the fingertips for the purpose of identification. The use of internal fingertip images for identification offers several potential benefits over traditional surface fingerprint scanners. One advantage is that internal fingertip images may be more resistant to spoofing and other forms of tampering since they capture the deeper layers of the skin that are not visible from the surface. This can make it more difficult for an unauthorized person to impersonate someone else using a fake fingerprint (Biggio, B., Akhtar, Z., Fumera, G., Marcialis, G. L. and Roli, F., 2012). Internal fingertip images captured using OCT offer a potential advantage in identification accuracy and reliability compared to surface fingerprint scanners. Factors like sweat, dirt, and wear can compromise the accuracy of surface scans, whereas OCT captures the deeper layers of the skin, minimizing these issues.

However, the complexity of OCT images and the need for sophisticated optics and advanced image processing techniques pose challenges in extracting relevant features. In addition, it is important to ensure that the dataset of OCT fingerprints used for training the deep learning model is diverse and representative of the population to be identified. Furthermore, OCT images cannot be distinguished by the human eye due to the slicing data in the area 4x4 mm around 1,000 images per scanning, and previous research has successfully utilized these images to distinguish between fake and real fingers based on internal fingertip characteristics. Thus, we aim to apply OCT images to deep learning models to achieve accurate identification results.

1.2 Objectives of the research

1.2.1 Develop and investigate a deep learning model to recognize human fingerprint images obtained through Optical Coherence Tomography (OCT).

1.2.2 Establish a streamlined workflow for the deep learning process applied to fingerprint identification systems using OCT imaging.

1.2.3 Distinguish and categorize true positives and false negatives from a dataset comprising 12 unique fingerprints.

1.3 Scope and Limitations of the Study

This research project studies machine learning applied for in-house OCT system by utilizing convolution neural networks. First, preparing data, we will collect OCT fingerprint images from 12 unique fingers. Each scan using the OCT machine in the 4x4 mm area captures approximately 1,000 images per finger, providing a diverse range of information about the internal fingerprint structure. These images exhibit variations in position, depth, and area, allowing for a comprehensive understanding of the fingerprint characteristics. With all 12 fingers scanned, the dataset comprises approximately 12,000 images, offering a substantial and rich collection of fingerprint data. This extensive dataset enables a detailed analysis and enhances the model's ability to accurately classify and identify fingerprint OCT images. However, due to the constraints of our graphics processing unit's memory. Specifically, we utilize a GTX1060 with a 6GB capacity, which, while robust, ensures efficient processing and avoids out-of-memory (OOM) issues. For the spec of in-house OCT, the working concept will be followed by (Liu, X., Zaki, F., Wang, Y., Huang, Q., Mei, X. and Wang, J., 2017); the system uses a near-infrared broadband light source (superluminescent diode (SLD), 800-840 nm central wavelength, and 60 nm bandwidth). The source of low-coherence light (SLD) emits light through a fiber optic Michelson-interferometer configuration, consisting of a sample arm and a reference arm. Lens 30 nm focal length had been used in sample arm to focus probe beam and receive the signals back from a finger. Light propagated back from the sample and reference is detected by a charge-coupled device (CCD) sensor. We obtain the OCT fingerprint images. Then the software-based solution gives us the grey-scale images with depth information. For the deep learning part, we be followed the design CNNs by LeCun, Bengio, et al., 2015, the related model, and utilize the pre-trained model to compare the efficiency with the result. We increase the performance by using the ensemble model for the combined multiple models, and then the system vote and average the results for the best outcome. The

evaluation accepts the appropriate model, for instance, the confusion matrix and F1-score. The model aims to classify and identify fingerprint OCT images with accuracy of AUC more than 0.98.

In this research, deep learning had been used to learn the images from captured internal fingertip images for prediction and identification. The scopes are listed as follows:

1.3.1 Data collected only from twelve unique user fingers.

1.3.2 This study was based on using CNN and ANN.

1.3.3 The study employed pre-trained deep learning models such as Inceptionv3, VGG16, Xception, and ResNet50 along with a custom model for feature extraction from the internal fingertip images.

1.3.4 The study incorporated a voting method for ensuring the robustness of the proposed security system.

1.4 Expected results

1.4.1 Identification system could predict unseen cross-sectional fingerprint oct image datasets with high accuracy.

1.4.2 Proposing an application of OCT machines as a security system by incorporating fingerprint information to apply system security.

CHAPTER II

LITERATURE REVIEWS

In this chapter, the working principles of optical coherence tomography (OCT) for fingerprint identification are explored. Various techniques proposed for capturing and processing OCT images of fingerprints are examined, along with the performance of these techniques in terms of accuracy, reliability, and security. Limitations or challenges associated with using OCT for fingerprint identification are also discussed, as well as potential future research directions in this area. Furthermore, the literature review includes a comparison of the results obtained using OCT and those acquired through traditional fingerprint scanners.

2.1 Fingerprint scanner

A fingerprint scanner is a tool used to collect and examine prints to identify or verify someone. Nowadays, many applications, including security systems, mobile devices, and attendance monitoring systems, use fingerprint scanners extensively. This traditional technology works by capturing an image of a surface fingertip and then analyzing the characteristics and pattern of the image for matching with the template. Standard technology generally has three types of sensors that are 1) capacitive sensor (Lee, J., Min, D., Kim, J. and Kim, W., 1999), 2) optical reflection (Soifer, V., Kotlyar, V., Khonina, S. and Skidanov, R., 1998), and 3) ultrasonic sensor (Lu, Y., Assaderagh, F., Daneman, M., Jiang, X., Lim, M., Li, X... 2016). Matching score for these technologies, the system will raise the score with a threshold called the matching score. If a score exceeds the threshold, the user can access the system as authentic; other systems will be rejected as an imposter.

2.1.1 capacitive sensor

Most smartphone secure technology utilizes a capacitive scanner; this type of scanner is named after the capacitive sensor. The scanner uses arrays of tiny capacitors for the working principle to collect the data, as shown in Figure 1.

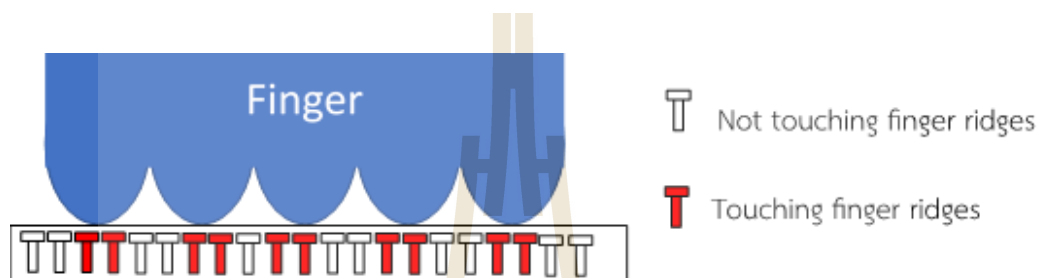


Figure 1 Schematic of working principle of capacitive scanner.

The sensor comprises a matrix of small capacitors used to measure the electrical charge on the surface of the finger. When a finger is placed on the sensor, the electrical charge on the surface of the finger changes the capacitance at each point on the sensor, creating a unique fingerprint image. Hence, the system will store the unique fingerprint for the database and compare it when users try to access it. Hackers cannot use a two-dimension image to fraud a system. So, more capacitive sensors, more security.

2.1.2 Optical scanner

The oldest technology for a scanner is optical type; this method uses algorithms to detect unique patterns from the finger surface, for instance, ridges or marks from a fingertip, as shown in Figure 2.

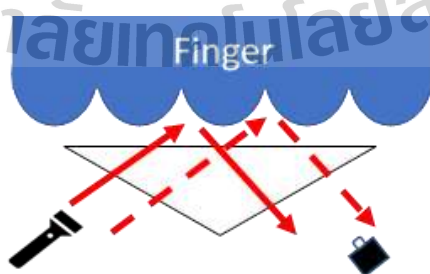


Figure 2 Schematic of working principle of an optical fingerprint scanner.

This type of scanner works by capturing the image of a fingerprint from the light source and a charge-coupled device (CCD) or a complementary metal-oxide-semiconductor (CMOS) sensor. The light emits to a surface fingertip then a pattern of ridges and valleys scatters the light in different directions. CCD or CMOS sensor will capture the light and convert it to an electrical signal. So, that could create an image matching the pre-existing template to identify and verify the individual. Not only optical scanner only captured two-dimension images but this sensor also depends on the quality of receiving sensor. That is mean good image quality will get more secure.

2.1.3 Ultrasonic sensor

An ultrasonic fingerprint scanner works by capturing an image of the fingerprint using ultrasonic waves. the equipment includes the ultrasonic transmitter and receiver. The transmitter will transmit the ultrasonic pulse incident to the fingertip. Some pulses are absorbed, and some are bounced by the receiver sensor as Figure 3.

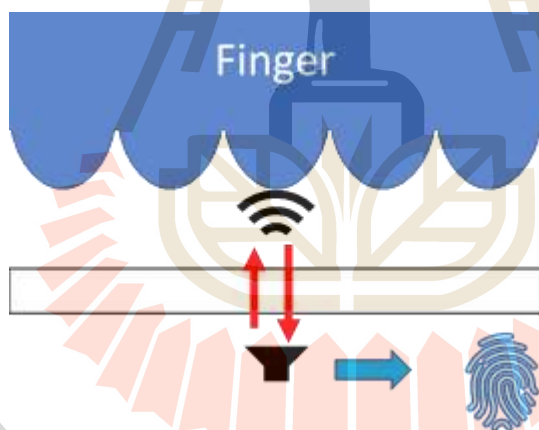


Figure 3 Schematic of working principle of an ultrasonic fingerprint scanner.

The sensor received the signal and then created a map of the fingerprint's ridges and valleys. The scanner then compares this map to a pre-existing template to determine a match and verify the identity of the individual.

In summary, conventional fingerprint sensors primarily rely on 2D information from the surface of the fingertip. The goal is to enhance security by integrating

in-house technology with new applications, which could provide a valuable alternative to system security.

2.2 Optical coherence tomography

Optical coherence tomography (OCT) technology, invented in the 1990s, is equipment that is equivalent to ultrasonography. Let's say the technology is analogous to the image captured by ultrasound imaging but has a much better resolution (Huang, D., Swanson, E. A., Lin, C. P., Schuman, J. S., Stinson, W. G., Chang, W., Hee, M. R., Flotte, T., Gregory, K. and Puliafito, C. A., 1991). Furthermore, OCT is a high-resolution, non-invasive optical imaging modality based on low-coherence interferometry or near-infrared light rather than possibly dangerous ionizing radiation, for detecting depth-resolved 2-dimension and 3-dimension images of highly scattering semi-transparent, e.g., biological tissue. For types of OCT, time domain optical coherence tomography (TD-OCT) and spectral-domain optical coherence tomography (SD-OCT) are the primary type of OCT (Fercher, A. F., Hitzenberger, C. K., Kamp, G. and El-Zaiat, S. Y., 1995). TD-OCT measures the temporal interference of a broad-band light beam; its imaging speed is significantly constrained by the need to mechanically scan its reference mirror to obtain a depth-sectioning image (Bouma, B. E. and Tearney, G. J., 2002). On the other hand, it analyzes interference in the light spectrum without scanning the reference mirror, significantly speeding up imaging (Fercher, A., Drexler, W., Hitzenberger, C. and Lasser, T., 2003). Various fields, including dermatology, ophthalmology, biology, aquatic toxicity, and material characterization, have already used OCT technology.

2.2.1 Michelson interferometry

The working principle of OCT is based on the Michelson interferometry. The light source emits the wave to the beam splitter. The incident wave will split into two paths. One half is reflected by a reference mirror. The other is reflected by the sample. The back-reflected beam will be interferenced with by the beam splitter then the detector receives the signal in Figure 4.

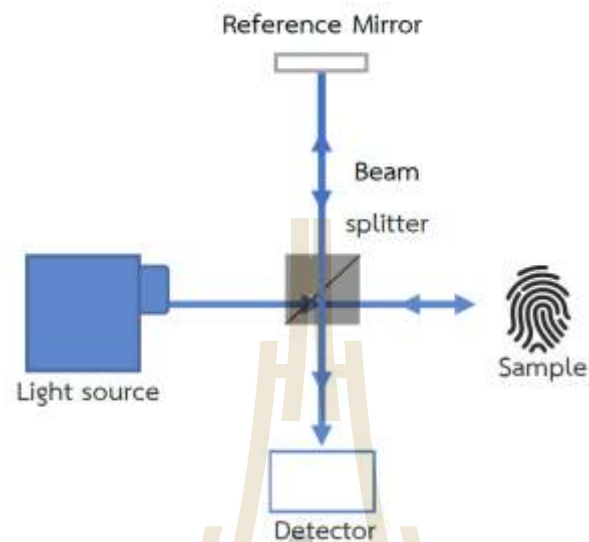


Figure 4 Schematic of Michelson interferometer.

Lastly, several depths in the biological tissue were backscattered from the sample beam that was reflected from a mirror. These images provide a detailed cross-sectional view. Accordingly, in-house software will convert the signal to images with depth information.

2.2.2 Signal processing for image generation

The acquired interferometric signals need to be processed to generate meaningful images. Fourier transformation is employed to convert the interference signals into depth-resolved A-scans. Signal dispersion, caused by variations in the refractive index, is a significant challenge in OCT imaging. Sophisticated algorithms and techniques have been developed to compensate for dispersion and improve image quality. Various data acquisition schemes, such as time-domain and frequency-domain OCT, have been explored to enhance imaging speed and sensitivity (Huang, D., Swanson, E. A., Lin, C. P., Schuman, J. S., Stinson, W. G., Chang, W., Hee, M. R., Flotte, T., Gregory, K. and Puliafito, C. A., 1991).

2.2.3 Data augmentation

Data augmentation involves making random adjustments to the dataset, which enhances its quality during training. This technique can mitigate overfitting, leading

to improved model performance. Moreover, data augmentation can expand the dataset by generating additional variations. Perez et al. demonstrated that image augmentation could boost the performance of deep learning models (Perez, L. and Wang, J., 2017). Figure 5 below illustrates an example of applying the augmentation technique to OCT fingerprint images.

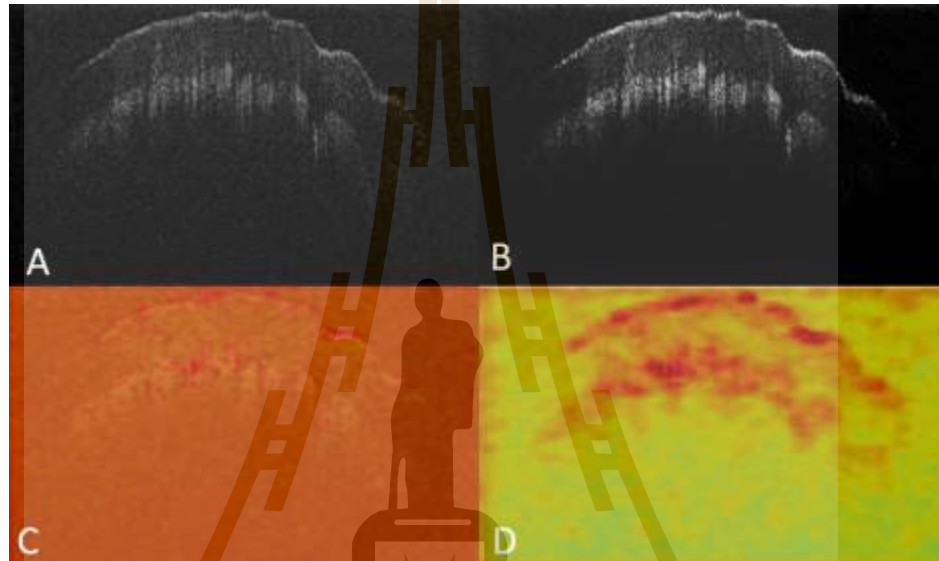


Figure 5 Image processing which A)raw data B) contrast adjust with noise filter (median filter) C) and D) Heat mapping.

Figure 5 demonstrates an example of adapting image processing techniques for OCT fingerprint images in a training dataset before training the model. Image A represents the raw data, while image B shows the processed image with adjusted noise (median filter) and contrast. Images C and D present heat maps to basic visualize how the networks extract features from the photos and gather information from various aspects. After image processing, OCT fingerprint images contain a rich of details. As the surface is densely packed with information, this area can be considered as a biomarker for fingerprints. (Almahdi, R. and Ragb, H., 2019).

2.2.3.1 Noise reduction technique

After the processing image from OCT system, the sparkle noise or granular appearance is the consequent of coherent light from using OCT. That's lead to

downgrading the quality of image and effect to the model's performance. The noise removal technique had been processed in the image for control the image quality. Noise removal technique will show in below:

- Median filter: a median filter has proven effective in enhancing image quality and improving the performance of models. The median filter is a non-linear spatial filtering technique that replaces each pixel value with the median value within its local neighborhood. This method effectively reduces the influence of speckle noise while preserving important image details and edges. By taking into account the statistical properties of the pixel neighborhood, the median filter successfully suppresses noise without causing significant blurring or loss of relevant image information (Zhu, Y. and Huang, C., 2012). The approximately of noise variance of the median filtering is below:

$$\sigma^2 = \frac{1}{4n^2(\bar{n})} \left(\frac{\sigma_i}{n + \frac{\pi}{2}} - 1 \right)^2 \cdot \frac{\pi}{2}$$

Where σ^2 is input noise power (the variance), n is the size of the median filtering kernel, $f(\bar{n})$ is the function of the noise density. And the noise variance of the average filtering is

$$\sigma^2 = \frac{1}{n} \sigma_i^2$$

- Rotating kernel transformation (RKT): RKT technique is an advanced method for noise removal and image enhancement. RKT employs a rotating kernel transformation to address noise-related issues. By applying a rotation to the kernel matrix, RKT introduces a unique transformation that effectively reduces noise while preserving important image details. The adaptive nature of the rotating kernel allows it to capture and suppress various types of noise, including random and structured patterns. The RKT technique has demonstrated promising results in enhancing image quality and

improving the performance of models, making it a valuable tool for noise reduction in a wide range of applications (Rogowska, J. and Brezinski, M. E., 2000).

2.2.4 Presentation attack

The fingerprint image from optical coherence tomography is called the OCT fingerprint. The stratum corneum (surface fingerprint), the epidermis, the internal fingerprint (papillary junction, sweat glands, and the dermis) will all have depth information, which implies the image will have this information (Chugh, T. and Jain, A. K., 2019b). The information had shown in Figure 6.

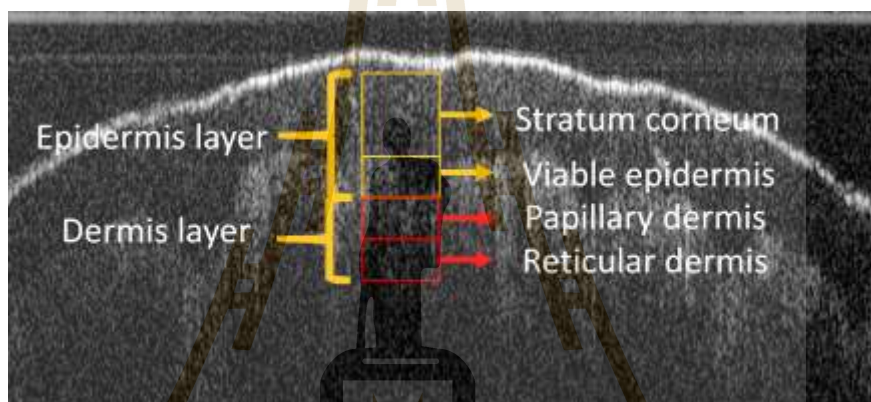


Figure 6 B-scan image with the dept information of fingertip.

According to the traditional fingerprint scanner that based on the surface of fingerprint, that made them easy to hack by presentation attack instruments (PAIs) with highly accurate finger surfaces. PAIs were created with inexpensive and easy-to-find materials, for instance, play-doh, gelatin, wood glue etc. These materials can use to produce an instrument to fraud into a security system, with success reported at around 70% (Chugh and Jain, 2019).

2.3 Deep learning

A branch of machine learning called deep learning (DL) uses a network of features to learn from the dataset's statistical information and can learn and decide by itself. The architecture of deep learning is based on artificial neural networks or has a functional design that resembles the organic neural network of the human brain (Goodfellow, I.,

Bengio, Y. and Courville, A., 2016). To create the models, DL typically divides the data into training and testing sets. To replicate the analysis of future data for the testing set, the training set's goal is to improve forecast data precision and fine-tune model parameters. Accordingly, DL is more practical than classic ML since it integrates "end-to-end" data learning of feature representations and model predictions (Figure 7). Deep neural networks are one type of trained computer model (DNN) (LeCun, Y., Bengio, Y. and Hinton, G., 2015).

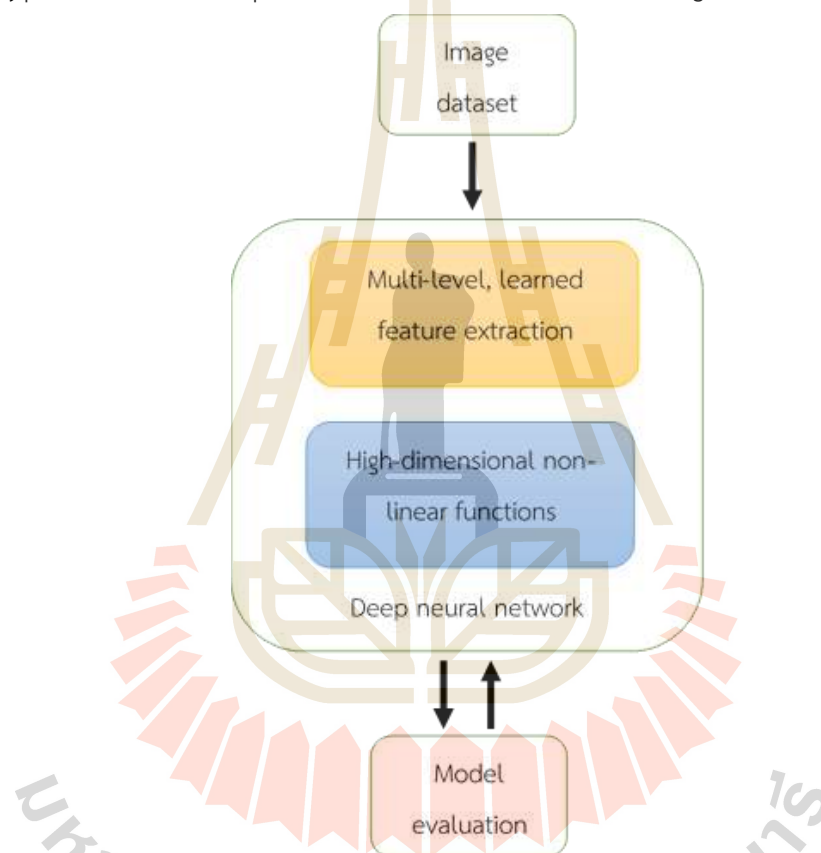


Figure 7 Deep learning: combining of feature extraction and prediction.

According to the literature review above, standard fingerprint identification technology is often based on FTIR and capacitive sensors and only takes into account the ridges or surfaces of the finger. Because of this, they were simple targets for presentation attack instruments (PAIs) with extremely precise finger surfaces. According to Akbari et al study of fingerprint recognition using OCT fingerprint photos, the algorithm for doing so is based on scanning and segmenting OCT images. They argue that increasing the automation

of fingerprint recognition will help this system succeed (Akbari, N., 2012). Moola et al. investigated the design utilizing an antiquated security method known as fingerprint matching for the security that used the fingerprint OCT. They used specialized hardware to capture the skin's lower layers, then they removed and translated the data for a compatible template. The study's integration of the matching OCT with the currently used standard fingerprint recognition has demonstrated the framework's capability. They contend that a system using OCT image matching has various advantages over existing technologies, including the capacity to improve existing technologies and boost matching system dependability (Moola, Y., Singh, A., Saith, E. and Akhoury, S., 2015). Numerous studies used OCT fingerprints in machine learning; Raja et al study, informs us about the transfer weight of AlexNet for two streams convolution neural network (TS-CNN). They train the network using 200 distinct fingerprints to show the various layered sub-surface OCT fingerprints. The model's performance was assessed using an equal error rate (EER), and as a consequence, they discovered an EER of 0.17 percent and results that were in line with the Neurotechnology Commercial-Off-The-Shelf (COTS) system.

In conclusion, presentation attacks can be avoided by using OCT images for fingerprint recognition. Unlike conventional fingerprint scanning techniques, which only record surface-level data, OCT scans to record the interior structure of the finger, which is more challenging to recreate. Therefore, using OCT pictures can offer a more secure means of authentication, lowering the chance of fabrication or impersonation. Additionally, the accuracy and resilience of the system can be further enhanced by the application of deep learning algorithms for fingerprint recognition using OCT images, offering a practical remedy for combating presentation attacks.

2.3.1 Convolutional neural networks

Convolutional Neural Networks (CNNs) are a class of deep learning algorithms that have revolutionized computer vision applications, including object detection, segmentation, identification, and classification. The term "convolutional" refers to the network's use of the mathematical operation called convolution, which involves processing input images with multiple layers of neurons. CNNs extract features from input

OCT fingerprint images with depth information, and feature extraction is the primary function of CNNs for prediction. Let's discuss the layers of a CNN:

The convolutional layer comprises an input image and a convolutional matrix operator. It functions by applying a set of filters to the image to extract essential features, such as edges, textures, and shapes. The outputs are combined to generate a feature map. In summary, a convolutional layer is a vital building block of a CNN, enabling the model to learn features and representations of the input data in a manner that is well-suited for image processing tasks. The weight matrix or convolution matrix is also referred to as a kernel or filter. This filter operates on the image data, starting at the edge and shifting to other pixels based on the set parameters. After processing the entire image, a new image is generated for the next layer. This layer reduces the dimension of the input image. An example is provided in Figure 8 below:

For the definitions in CNN that should concern will describe in below.

- Convolution (conv): Explain how convolutional layers extract local features by applying filters to input data.
- Pooling (pool): Describe the process of downsampling input data to reduce spatial dimensions while preserving important features.
 - Max Pooling: The max pooling operation selects the maximum value within each pooling window and discards the rest. It reduces the spatial dimensions of the feature map while retaining the most prominent features.
 - Average Pooling: Average pooling calculates the average value within each pooling window. It helps in downsampling the feature map while maintaining an overall representation of the input data.
 - Global Pooling: Global pooling performs pooling across the entire spatial dimensions of the feature map, resulting in a single value per channel. It is often used in the final layers of CNNs for spatial summarization before the fully connected layers.

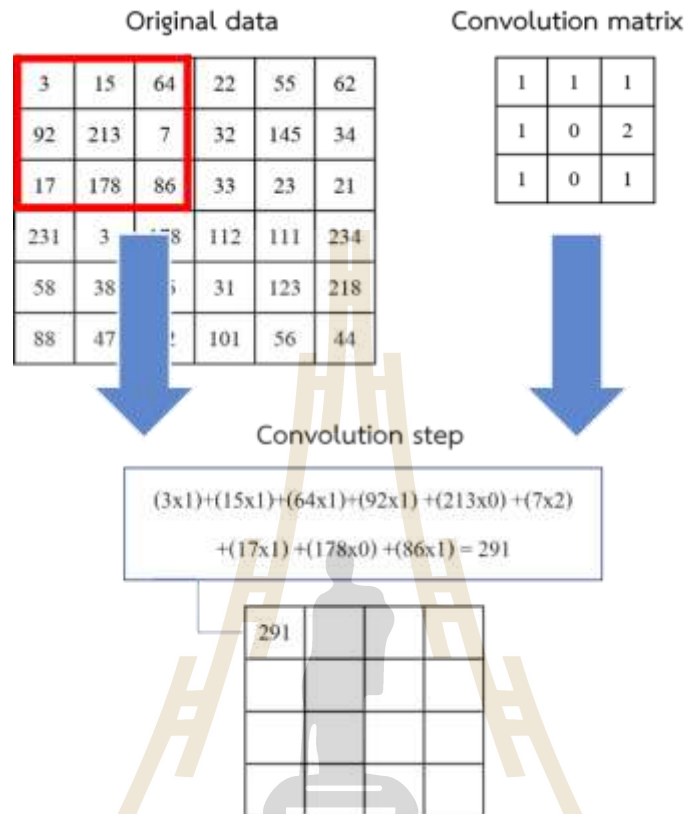


Figure 8 Convolution calculation.

The convolution operation typically results in reduced size of the output image. However, if it is desired to maintain the size of the original data, padding can be applied before the convolution process. Padding is a technique that involves adding zeros around the edges of the original image, as illustrated in Figure 9. This process helps preserve the spatial dimensions of the input image, ensuring that the output image retains the same size as the original.

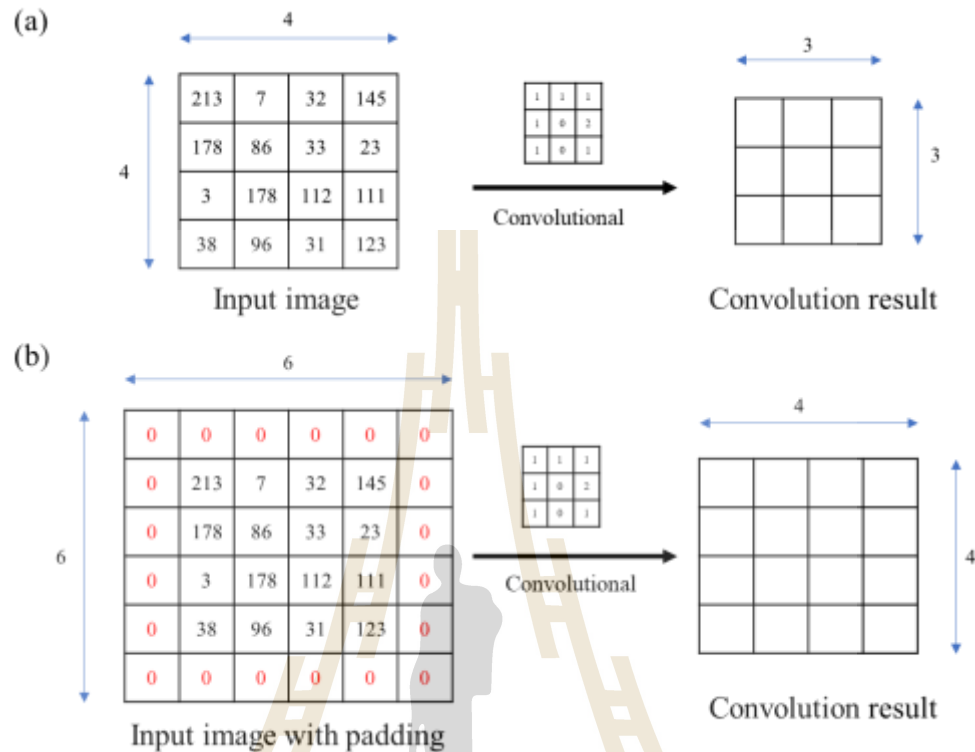


Figure 9 Convolution operation (a) zero padding (b) padding is 1.

Pooling layers come in various forms for constructing CNN architectures, but they all share the common goal of progressively reducing the network's spatial dimensions. This reduction lowers the number of parameters and simplifies the overall computation of the network. In this project, both individual models and transfer learning approaches primarily employ the Max-pooling technique. With the max-pooling method, the pixel value of the resulting image is determined by selecting the highest value within each local group. Figure 10 provides an example of the max-pooling process in action.

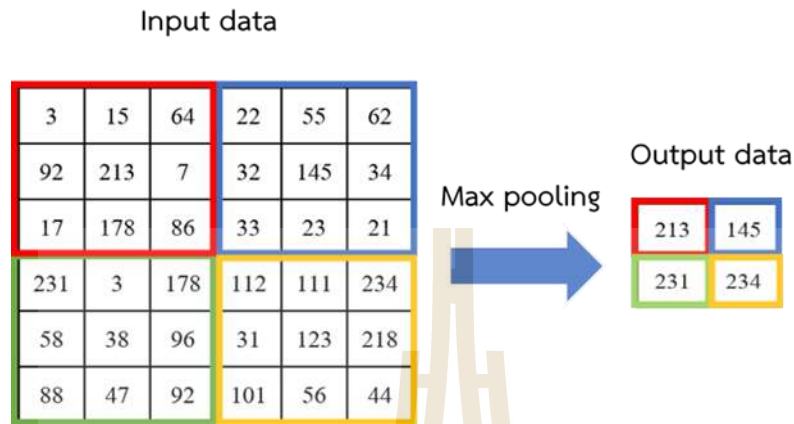


Figure 10 Max pooling.

The Flatten layer transforms a three-dimensional layer into a one-dimensional vector, which can then be fed into a fully connected layer for classification. For instance, a 5x5x2 layer would be converted into a vector of size 50. The resulting outputs represent the probabilities of each category, which are used to classify the input image.

2.3.2 Artificial neural networks

Artificial Neural Networks serve as the foundation of deep learning, enabling complex pattern recognition, decision-making, and prediction tasks. By mimicking the structure and function of biological neural networks, subunits called neurons or nodes. Each neuron associated with which weight factors $(w_1, w_2, w_3, \dots, w_n)$ to determine the importance of input values $(x_1, x_2, x_3, \dots, x_n)$ as shown in figure 11. Each neuron connection is assigned a weight, and the input signals are multiplied by these weights. Additionally, a bias term is incorporated to adjust the output signal, which is determined by the discrete steps based on the summation of the input signals. The activation of the output neuron can be mathematically expressed as follows:

$$\text{Output of neuron} = \text{Activation Function} \left(\sum_{i=1}^n x_i w_{ij} - \theta_j \right)$$

Where n is the number of input connections into unit j .

w_{ij} is the weight of the connection between unit i of the previous layer and unit j .

θ_j is the bias that adjusts the output signal to provide neuron with a trainable constant value.

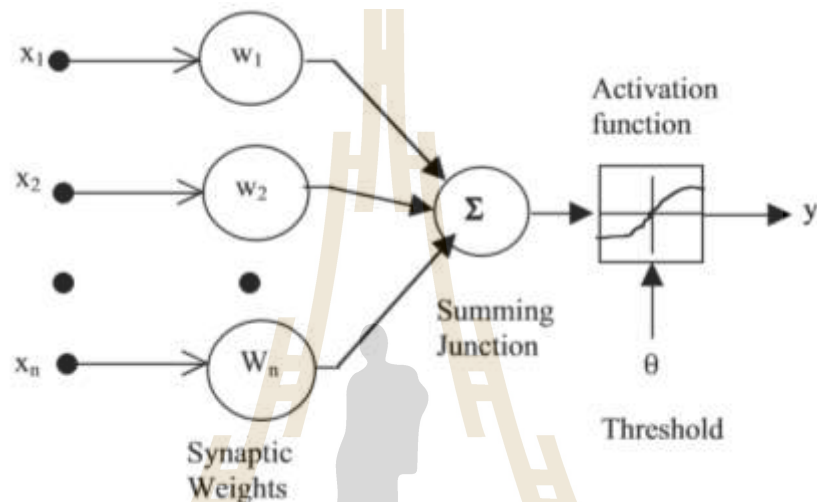


Figure 11 Basic artificial neuron (Zilouchian, A. and Jamshidi, M., 2001).

The activation function plays a crucial role in converting the computed value from the weighted sum of a neural network's structure. Its primary function is to transform the activation level of a neuron, denoted as y , into an output value. Various activation functions are commonly encountered in neural networks, including:

- **Linear Function:** returns the same value that was used as its argument.

$$f(x) = x, \infty \leq f(x) \leq \infty$$

- **Sigmoidal function:** returns the value input and squeeze the remaining values 0 to 1.

$$f(x) = \frac{1}{1 + e^{-x}}, 0 \leq f(x) \leq 1$$

- **Hyperbolic tangent function:** resemble sigmoidal function. The difference is that the negative inputs will be mapped negative, and the zero inputs will be mapped near zero.

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}, -1 \leq f(x) \leq 1$$

- **Rectified Linear Unit (ReLU):** ReLU activation sets the output to zero for negative inputs and leaves positive inputs unchanged. It offers faster convergence during training and helps mitigate the vanishing gradient problem (Agarap, A. F., 2018).

$$\text{ReLU}(x) = \max(0, x)$$

For the definitions in ANN that should concern will describe in below.

- **Fully Connected (FC) Layer:** Define the layer in which all neurons are connected to every neuron in the previous and following layers.
- **Softmax:** Explain the activation function often used in the output layer for multiclass classification problems.
- **Dropout:** Define the regularization technique that randomly sets a portion of neurons to zero during training to prevent overfitting.

From these knowledge, neural network could be combined from CNN and ANN design a powerful architecture for training models, we can construct a robust and adaptable framework. This foundation enables us to delve into the exciting realm of transfer learning, where pre-trained models can be leveraged to enhance performance on new tasks with limited data (Zilouchian, A. and Jamshidi, M., 2001).

2.3.2 Transfer learning

Transfer learning is a technique that leverages pre-trained models to adapt to new tasks with limited data. This technique has been applied to a variety of jobs, including speech recognition, natural language processing, and computer vision. It has been demonstrated to be successful in tasks where labeled data is few or where obtaining labeled data is expensive. Transfer learning can perform better than training a model from start on a new task because it makes use of the knowledge acquired from one assignment, by fine-tuning a pre-trained model on a dataset of OCT images of fingerprints, the diagram of the concept of transfer learning showed in figure 12. The primary goal is to efficiently

capture the features of the fingerprints and enhance the accuracy of the identification process. (Yosinski, J., Clune, J., Bengio, Y. and Lipson, H., 2014).

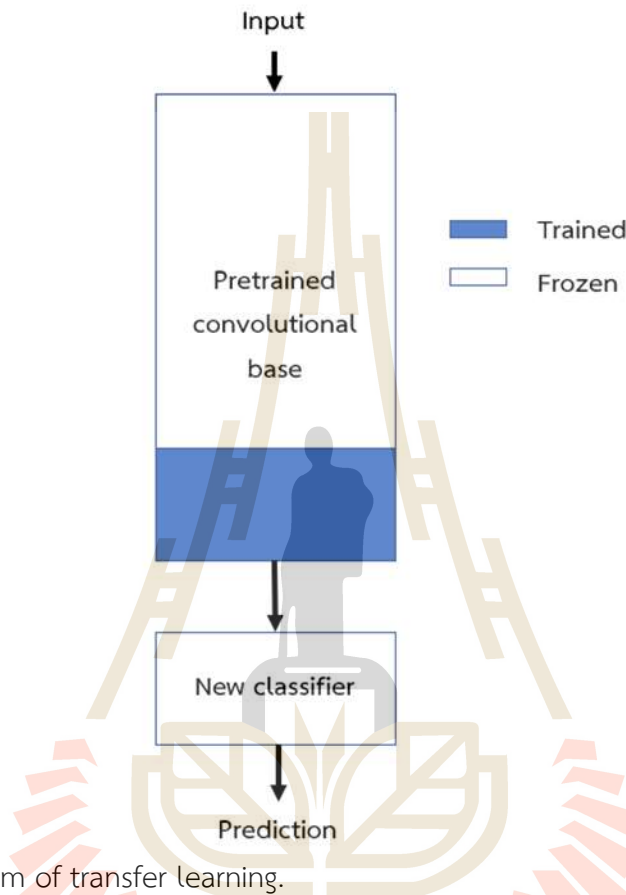


Figure 12 diagram of transfer learning.

In this study, different pre-trained models, such as ResNet50, Xception, VGG16, and InceptionV3, are explored and their performance on the task of identifying human fingerprints using OCT images is compared. The focus will now shift to a closer examination of the selected pre-trained models for this study.

2.3.2.1 ResNet50

Starting with the pre-trained ResNet50 model. A deep residual network called ResNet50 was developed and trained on the ImageNet dataset, which consists of millions of photos of various objects. The enormous dataset from which our pre-trained model has previously learned rich features can be used as the foundation

for our fingerprint classification challenge, the pre-trained ResNet50 model will be fine-tuned by training it on a smaller dataset of OCT images of human fingerprints. This fine-tuning process will update the weights of the pre-trained model. The structure of ResNet50 will show in Figure 13 (He, K., Zhang, X., Ren, S. and Sun, J., 2016).

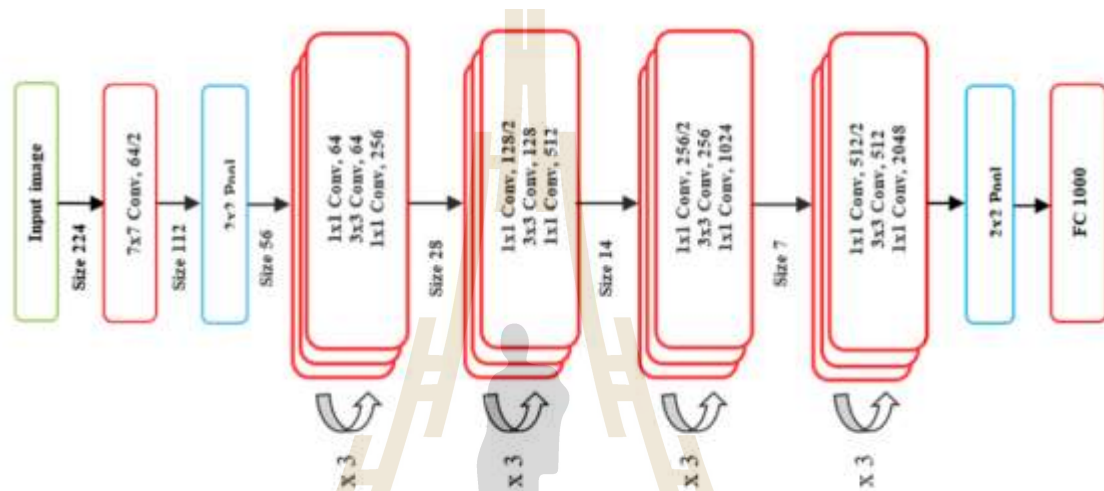


Figure 13 Resnet-50 model architecture (Ridha Ilyas, B., Beladgham, M., Merit, K. and taleb-ahmed, A., 2019).

2.3.2.2 Xception

A deep convolutional neural network model termed the Xception architecture has been pre-trained using the ImageNet dataset, a sizable image categorization dataset. A distinct but related job, such as identifying fingerprints using optical coherence tomography (OCT) pictures, can be solved using this model as a framework for transfer learning. In transfer learning, the pre-trained model is adjusted for the new job by adding fresh data in this case, OCT images to the weights of its layers. The architecture of Xception is made up of several stacked blocks of pointwise and depthwise separable convolution layers. The pointwise convolution layers capture the channel-wise information, while the depthwise separable convolution layers collect the spatial information. The Xception model's output can be linked to a fully connected layer that has as many neurons as classes in the target task—in this case, classes for various

fingerprints—in the target task. The output of the fully connected layer is used to forecast an input image's class. When the OCT photos are used to train the optimized Xception model from beginning to end, the accuracy of the task of fingerprint identification is increased in comparison to starting from scratch. The structure of Xception will show in Figure 14 (Chollet, F., 2017).

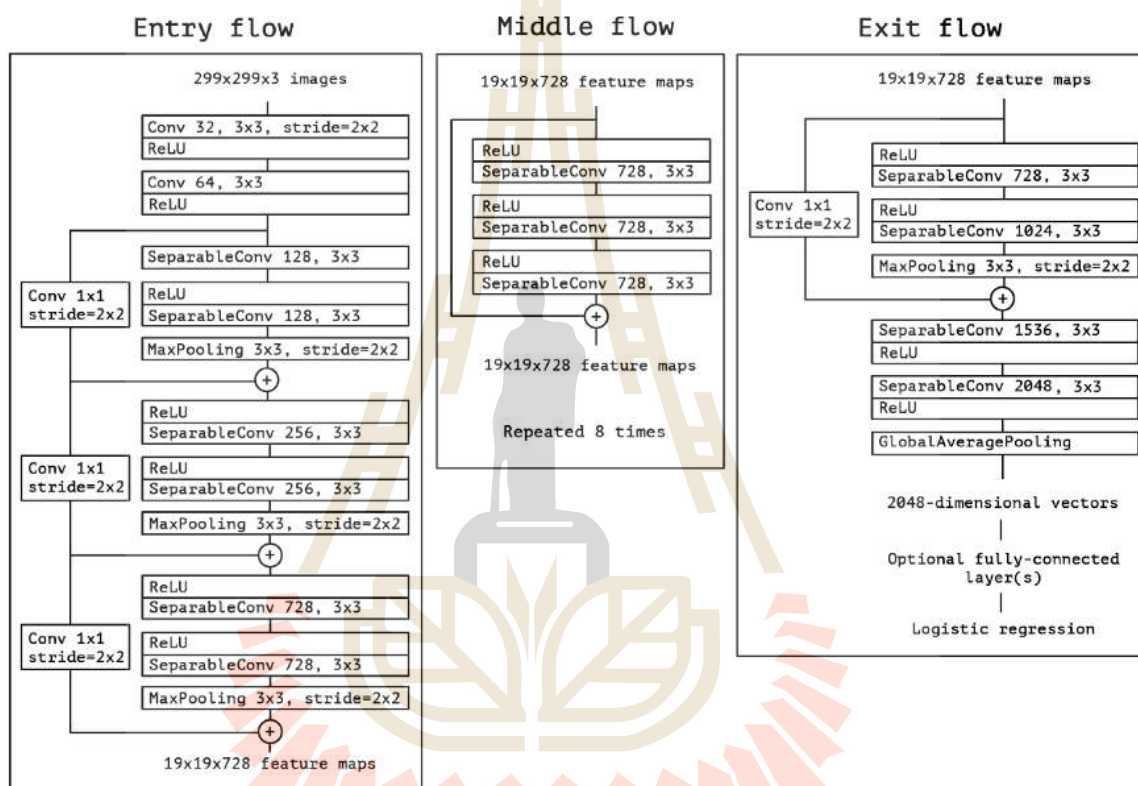


Figure 14 Xception neural network architecture diagrams of the model (Chollet, F., 2017).

2.3.2.3 VGG16

The VGG16 architecture is a deep convolutional neural network model that was previously trained using the ImageNet dataset, a sizable image classification dataset. In transfer learning, the pre-trained model is adjusted for the new job by adding data in this case, OCT images to the weights of its layers. VGG16's architecture is made up of several convolution and pooling layer stacks that are then followed by a few fully connected layers. While the pooling layers minimize the spatial dimension of the feature maps and preserve the most crucial data, the convolution layers

collect both spatial and channel-wise information. Using the input image as a basis, predictions are made using the fully connected layers. A fully connected layer with several neurons equal to the classes in the target task, in this case, the number of classes for various fingerprints, can be connected to the output of the VGG16 model. The output of the fully connected layer is used to forecast an input image's class. The accuracy of the fingerprint identification task will therefore be increased when the tweaked VGG16 model is trained end-to-end utilizing the OCT images as opposed to starting from scratch. The structure of VGG-16 will show in Figure 15 (Simonyan, K. and Zisserman, A., 2015).

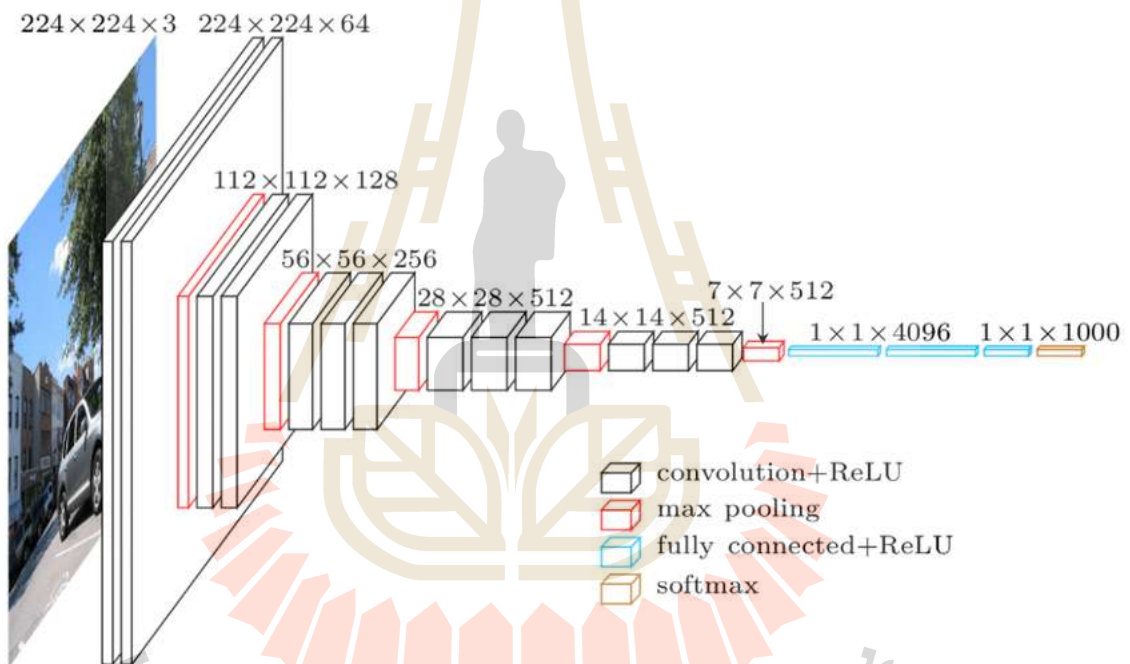


Figure 15 VGG16 neural network architecture diagrams of the model (Loukadakis, M., Cano, J. and O'Boyle, M., 2018).

2.3.2.4 InceptionV3

A deep convolutional neural network called InceptionV3 architecture was trained using ImageNet, a sizable image dataset. The pre-trained InceptionV3 network might be used in this project as a fixed feature extractor with a custom classifier built on top or as a feature extractor with variable features. The InceptionV3 architecture is made up of many Inception modules, each of which is made up of parallel convolutional and pooling layers, followed by a concatenation layer. The

pooling layer aids in reducing the spatial dimension of the feature map, while the parallel convolutional layers enable the network to learn multi-scale characteristics from the input image. The network's final dense layer is taught to produce class probabilities, which might then be retrained for the identification job using the fingerprint dataset. By fine-tuning the pre-trained weights of the network on the fingerprint dataset, it is possible to increase network performance by customizing it for the given purpose. The structure of inceptionV3 will show in Figure 16 (Szegedy, Christian, Vanhoucke, Vincent, Ioffe, Sergey, Shlens, Jonathon and Wojna, Zbigniew, 2016).

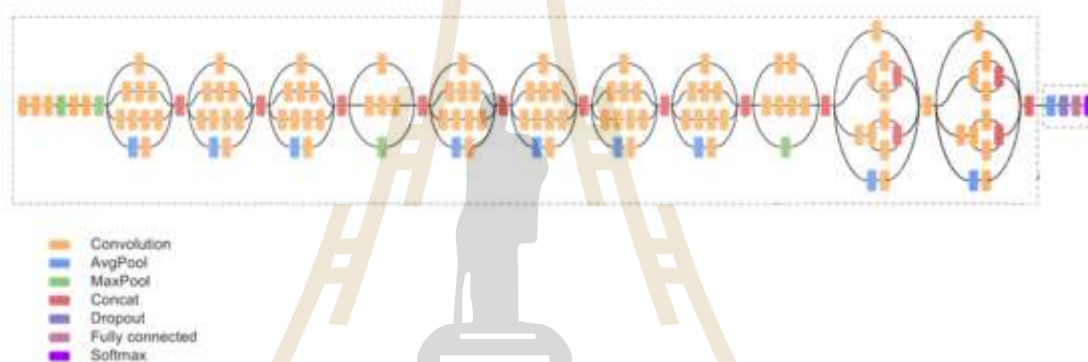


Figure 16 InceptionV3 neural network architecture diagrams of the model (Szegedy, Christian, Vanhoucke, Vincent, Ioffe, Sergey, Shlens, Jon and Wojna, Zbigniew, 2016).

In conclusion, the objective of this study is to apply deep learning and transfer learning to recognize human fingerprints by OCT images. A small dataset of OCT photos will be utilized to fine-tune the pre-trained ResNet50 model, and the resulting model will be used to categorize fingerprints in fresh, unused OCT images. Subsequently, multiple trained models will be combined to achieve the best prediction. This approach leverages the strengths of each model, enhancing overall performance and increasing the likelihood of obtaining accurate predictions.

2.3.3 Ensemble model

To achieve more reliable and accurate predictions. An ensemble model is suggested, a powerful technique to improve predictive performance. This approach combines the individual model and transfer learning mentioned earlier into an ensemble

model using a weighted average or another mathematical function, producing results as a single model. The ensemble method is typically applied in situations where a high level of accuracy is required due to data limitations and constrained processing resources. There are several ensemble techniques commonly used in practice, including but not limited to:

2.3.3.1 Bagging: Bagging (Bootstrap Aggregating) generates multiple subsets of the original dataset through bootstrap sampling, trains individual models on each subset, and combines their predictions through averaging or voting. This technique helps reduce model variance and improve stability as shown in Figure 17.

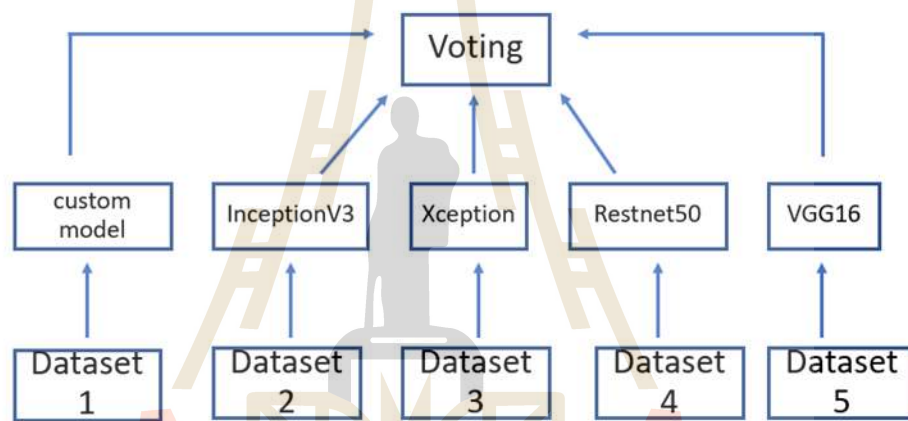


Figure 17 Bagging technique schematic.

2.3.3.2 Voting: A voting ensemble classifier is used, wherein each model provides a prediction, and the final prediction is determined by a majority vote. For example, if five models predict class A, class B, and class A respectively, class A would be the final prediction. In this case, there are 12 classes representing each unique finger. The diagram of the voting method is illustrated in Figure 18

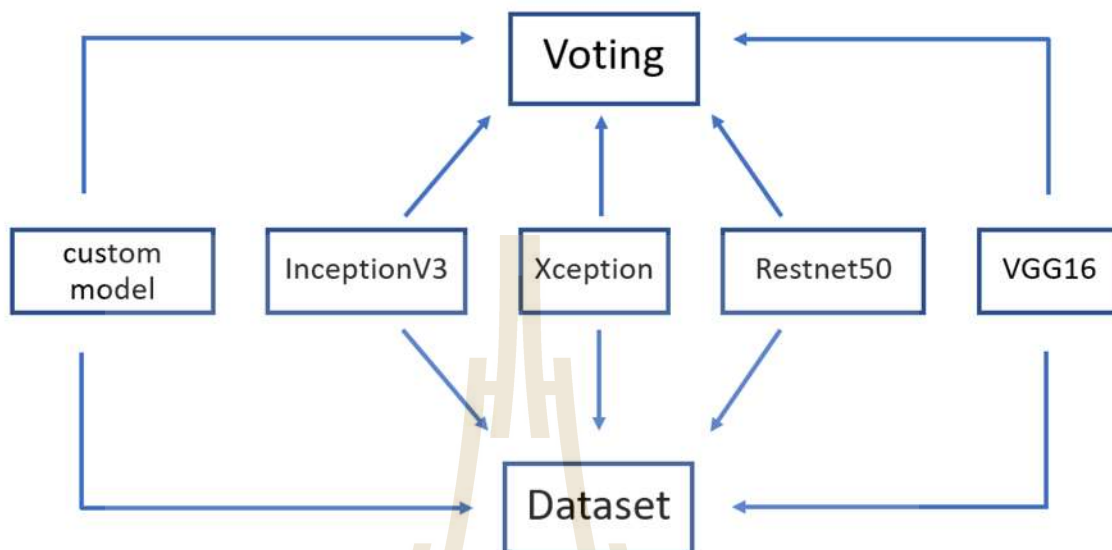


Figure 18 Voting technique schematic.

2.3.3.3 Boosting: Boosting is a type of machine learning that trains weak models one after the other to make a strong predictive model as shown in figure 19. It aims to improve how well models work by putting more attention on samples that were wrongly labeled before. Each time the process is repeated, the next model is taught to pay more attention to these wrongly classified samples, which "boosts" their importance. By combining the predictions of all the models, usually using a weighted voting system, boosting can make a powerful ensemble model that is great at handling complex patterns and getting high accuracy.

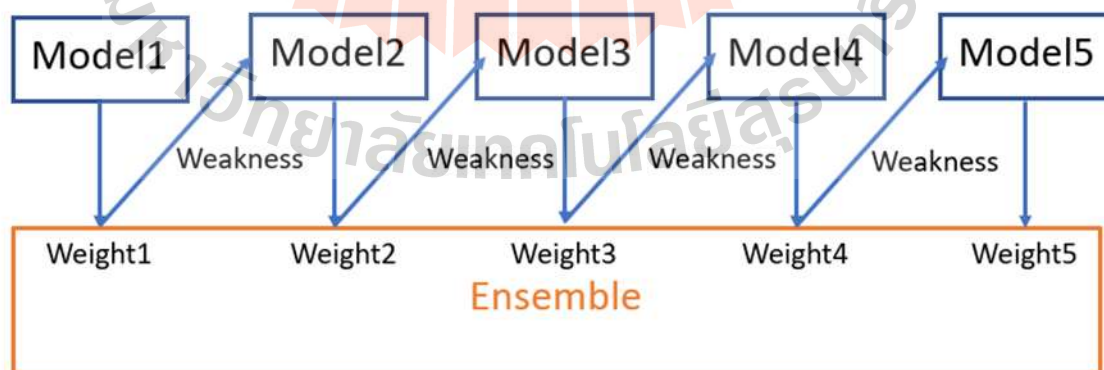


Figure 19 Boosting technique schematic.

2.3.4 Performance evaluation

Model evaluation is a crucial step in deep learning to measure the effectiveness of a trained model. This makes it easier to assess how effectively the model generalizes to unseen data. Accuracy is a popular performance statistic in deep learning that compares the proportion of accurate predictions a model makes to all other possible predictions. The performance of the model may also be assessed using additional metrics including ROC curve, precision, recall, F1-score, and confusion matrix. It is crucial to properly assess the model's performance because doing so might point out areas for improvement and provide guidance for future work (Kulkarni, A., Chong, D. and Batarseh, F., 2020; Li, Q., Yang, Y., Guo, Y., Li, W., Liu, Y., Liu, H. and Kang, Y., 2021).

The confusion matrix in Table 1 represents the count from the predicted and actual values of each class. It is a significant table that may be used to gauge how effective categorization models are.

Table 1 Confusion matrix.

| Predicted \ Actual | Actual: Yes | Actual: No |
|--------------------|---------------------|---------------------|
| Predicted: Yes | True Positive (TP) | False Positive (FP) |
| Predicted: No | False Negative (FN) | True Negative (TN) |

Where True Positives (TP): Predicted as correctly on event values.

True Negatives (TN): Predicted as correctly on no-event values.

False Positives (FP): Predicted as incorrectly on event values.

False Negatives (FN): Predicted as incorrectly on no-event values.

From the confusion matrix table, the data inside can be utilized to calculate various performance metrics, such as accuracy, precision, recall, and F1 score. These metrics provide a comprehensive assessment of the classification model's performance, helping to identify its strengths and weaknesses. It is widely used for classification.

Accuracy is the most common for evaluating performance, which is a measure of the accuracy of the model by considering all classes. Precision represents the accuracy of the model that predicts positive values. The recall represents how accurate the model is for correctly predicting positive classes. And F1 score is a value that calculates from the weighted harmonic mean between the precision and the recall. That four values could be written as follows.

- Accuracy is the proportion of correct predictions.

$$\text{Accuracy} = \frac{\sum \text{True Positive} + \sum \text{True Negative}}{\sum \text{Total Population}}$$

- Precision measures how accurate the predictions are.

$$\text{Precision} = \frac{\sum \text{True Positive}}{\sum \text{Predicted condition positive}}$$

- Recall measures how well that finds all the positives.

$$\text{Recall} = \frac{\sum \text{True Positive}}{\sum \text{Condition Positive}}$$

- F₁ score is the harmonic mean of the precision and recall.

$$\text{F}_1 \text{ score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

In addition, multi-classification can be visualized, and the performance is measured using the AUC-ROC curve for the variety of thresholds. ROC (Receiver Operating Characteristics) curve is a probability curve, and AUC (Area under the curve) represents the degree or measure of separability. They will determine how a model can discriminate between classes: Higher AUC, a better model. The ROC curve plots the true positive rate (TPR) or sensitivity on the y-axis and the false positive rate (FPR) on the x-axis (Figure 20).

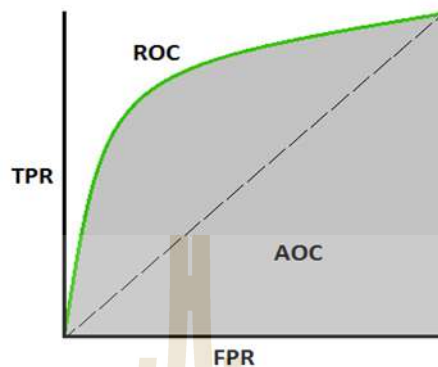


Figure 20 AUC-ROC curve.

However, it is important to note that the AUC-ROC curve provides insights into the overall performance of the model but does not explicitly address potential fitting problems, such as overfitting or underfitting. These fitting issues can impact the model's ability to generalize to unseen data and may affect its performance in real-world scenarios.

- **Underfitting:** Underfitting occurs when a model fails to capture the underlying patterns and relationships in the data, resulting in poor performance and low accuracy. Underfitting, in the context of AUC-ROC curve analysis, is a curve that doesn't distinguish between classes well enough, which leads to a low AUC score. This means that the model has trouble telling the difference between positive and negative situations. As a result, the model's classification performance isn't as good as it could be. Underfitting happens when the model is too simple or doesn't have enough complexity to capture the complexities of the data. This makes the model's representation too simple.
- **Overfitting:** Overfitting, on the other hand, arises when a model becomes excessively tailored to the training data, capturing noise and irrelevant patterns. In the AUC-ROC curve analysis, overfitting is reflected by a curve that exhibits high sensitivity and specificity on the training data but fails to generalize well to unseen data. As a result, the AUC score may appear high when evaluated on the training set, but it significantly drops when applied to new, unseen instances. Overfitting can occur when the

model is overly complex, capturing noise and specificities unique to the training set, rather than learning the general patterns of the underlying data distribution.

2.4 Explainable artificial intelligence

Explainable artificial intelligence (XAI) has emerged as an essential component of modern AI systems, particularly in the context of image classification. The need for interpretability and trust in AI models has driven research into techniques that facilitate better understanding and communication of model decisions (Barredo Arrieta, A., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A... 2020; Gilpin, L. H., Bau, D., Yuan, B. Z., Bajwa, A., Specter, M. and Kagal, L., 2018). In this literature review, the primary focus is on prominent XAI methods, with special attention given to SHapley Additive exPlanations (SHAP) for image classification tasks. The strengths and limitations of the SHAP approach are discussed to provide a comprehensive understanding of its applicability and potential challenges in the context of image classification.

2.4.1 SHapley Additive exPlanations (SHAP)

SHAP, developed by Lundberg and Lee (2017), is a model-independent method that produces explanations for specific predictions by attributing importance values to each feature using concepts from cooperative game theory (Lundberg, S. M. and Lee, S.-I., 2017). SHAP has been applied to various image classification tasks, demonstrating its effectiveness in producing explanations that are understandable to humans (Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D. and Batra, D., 2017). This method has been employed in multiple sectors, such as medical image analysis (Altmann, A., Toloşi, L., Sander, O. and Lengauer, T., 2010), remote sensing (Lundberg, S. M., Erion, G. G. and Lee, S.-I., 2018), and object recognition (Zhou, B., Khosla, A., Lapedriza, A., Oliva, A. and Torralba, A., 2016). SHAP has shown the ability to provide comprehensible classifications for complex models in each of these applications, assisting end users and domain experts in understanding and trusting the AI system's decisions (Simonyan, K., Vedaldi, A. and Zisserman, A., 2013).

Despite its benefits, SHAP has some limitations. For example, it can be computationally expensive for large-scale datasets or complex models, which may hinder real-time explanations (Molnar, C., 2020). Moreover, the quality of explanations may vary depending on the choice of the explainer method and the specific model being explained (Lundberg, S. M., Erion, G., Chen, H., DeGrave, A., Prutkin, J. M., Nair, B.... 2020).

SHAP is a popular XAI technique for image classification that offers interpretable and reliable explanations for sophisticated AI models. While SHAP has limitations, its effectiveness in various domains demonstrates its potential to enhance model decision comprehension and communication (Adadi, A. and Berrada, M., 2018). Future research should continue to address the constraints of SHAP, develop innovative XAI methodologies, and investigate their applicability in real-world situations (Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F. and Pedreschi, D., 2018).

2.4.2 LIME (Local Interpretable Model-agnostic Explanations)

LIME, proposed by Ribeiro et al. (2016), is another widely used model-agnostic interpretability method that aims to provide local explanations for individual predictions. This technique generates explanations by approximating the complex model's behavior with a locally interpretable model. LIME has been successfully applied to various domains, including image classification, and healthcare. For image classification tasks, LIME has been used to highlight the important regions or features in an image that contributed to a specific prediction. This approach aids in understanding the decision-making process of black-box models and facilitates human interpretability (Ribeiro, M. T., Singh, S. and Guestrin, C., 2016).

2.4.3 Grad-CAM (Gradient-weighted Class Activation Mapping)

Grad-CAM, proposed by Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2017), is an interpretability method that gives visual explanations for convolutional neural networks (CNNs) by highlighting the parts of an image that contributed most to the model's prediction. This method uses the gradient information from the CNN's last convolutional layer to make a class activation map that shows how important different parts of the image are. Grad-CAM has been used successfully for many

computer vision tasks, such as finding objects, dividing up images, and answering visual questions. By showing the important areas on a map, Grad-CAM gives useful information about how CNNs make predictions and helps people understand how decisions are made. It gives explanations that humans can understand by pointing out the parts of the image that affected the model's output. Grad-CAM is a flexible interpretability method because it can be used with both pre-trained and custom CNN architectures (Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D. and Batra, D., 2017).



CHAPTER III

METHODOLOGY

In this chapter, the methodology employed to implement the proposed solution for identifying human fingerprints using optical coherence tomography (OCT) images through deep learning is presented. The methodology encompasses the pre-processing of OCT images, selection and fine-tuning of pre-trained models such as ResNet50, Xception, VGG16, and InceptionV3, as well as the assessment of model performance using various metrics, including accuracy, precision, recall, and F1 score. Moreover, the chapter outlines the experimental setup, which covers the dataset utilized for training and testing, the hardware and software configurations, and the implementation specifics of the models.

3.1 Preparing data

In the data preparation step of the project, the left and right random fingers refer to twelve different people were captured using optical coherence tomography (OCT) imaging (Figure 22). An OCT machine was used to capture fingerprint images within a 4x4 mm area. Each scan generated approximately 1,000 grayscale images per finger, highlighting the variations in position, depth, and area of the internal fingerprint structure. With all 12 fingers scanned, the dataset comprised approximately 12,000 images, forming a comprehensive and diverse collection of fingerprint data for subsequent analysis and model development. Consequently, finger was captured in approximately 1,000 grayscale images for 3 times, providing depth information on the finger's surface and internal structure (Figure 21). Hence, in the dataset collection, the size of raw dataset about 3,000 images per class. This novel approach of using OCT imaging provided additional information about the finger beyond just the external surface and ridges of the fingerprint. The images obtained from OCT are often referred to as "layers," which can be used to

create a cross-sectional of 3D representation of the finger. This unique characteristic of the OCT-captured images provides a potentially more reliable and secure method of identifying individuals as it includes both external surface fingerprint data and internal information about the finger, such as sweat glands. Therefore, the proposed system has the potential to be superior to conventional fingerprint identification methods, and the collected dataset can be used for training and evaluating deep learning models.

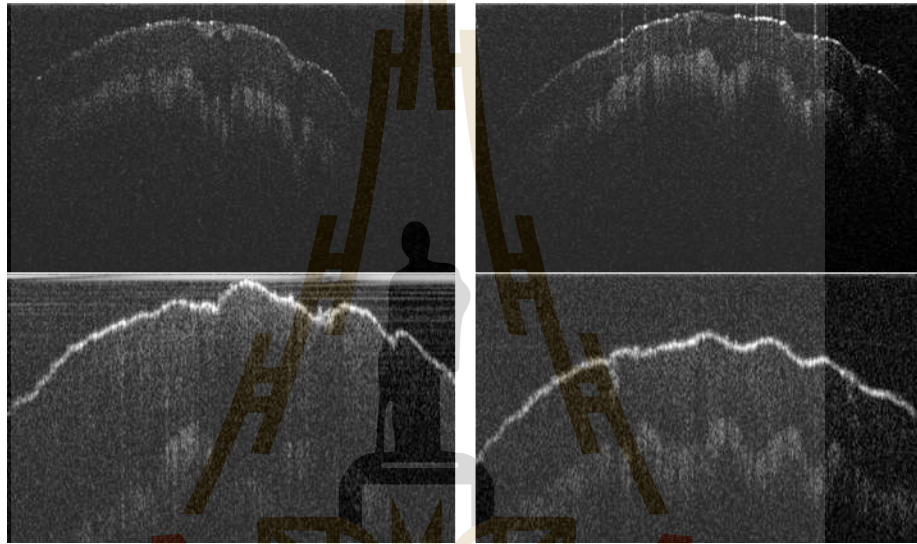


Figure 21 Example of finger OCT captured.



Figure 22 In-house optical coherence tomography.

In addition to collecting and organizing the OCT data, data augmentation, and processing techniques will be applied to enhance the quality and diversity of the dataset. Various image processing techniques, such as noise reduction, brightness adjustment, and contrast enhancement, will be employed to generate new images from the original OCT data. This process will increase the variety of the dataset, which is essential for the deep learning model's performance. Furthermore, heat mapping will be utilized to understand which regions of the images the model is using for classification. By visualizing these regions, insights can be gained into the fingerprint features the model is using for identification, which can be beneficial for further improving the model's performance.

In the data preparation phase, we captured multiple samples of each finger for the training set. To enhance the challenge and improve the model's capability, we applied digital processing techniques to these captured images. Subsequently, we randomly selected about 1,000 for a training set, with 900 images from each image collection and an additional 100 random images from the augmentation process. This ensured that each class in the training set had around 900 randomized images and 100 digitally processed images, resulting in a total of around 1000 images per class or finger. For 12 classes, the number of training data sets is 12,108 internal fingertip images. This approach allowed us to create a diverse and comprehensive training dataset that encompassed variations and challenges commonly encountered in fingerprint identification tasks.

3.2 Model Architecture and Selection

In this study, the methodology for developing the model architecture involved two approaches: building custom models and using pre-trained transfer learning models. As described in the literature review (section 2.3.2), the individual models that were built with shallow layers, such as pooling, batch normalization, dropout, and CNNs, did not perform well in terms of accuracy and generalization. Hence, pre-trained transfer learning models were utilized, including ResNet50, Xception, VGG16, and InceptionV3, which were selected based on their high performance in previous image classification tests and their availability in popular deep learning frameworks like Keras. By leveraging features learned

from a diverse and large dataset like ImageNet, transfer learning allowed the model to learn more efficiently from the limited fingerprint dataset available, thereby improving the model's accuracy and generalization capabilities for the specific fingerprint recognition challenge.

The custom CNN model was created using specific parameters: a convolutional layer with a kernel size of 3x3 for extracting 128 features. At this layer, the activation function ReLU was utilized (Agarap, A. F., 2018). The pooling layer employed max-pooling to obtain essential 2x2 information. During training, 500 epochs, a batch size of 128, and a learning rate of 0.0001 were used. The architecture of the custom model is illustrated in Figure 23.

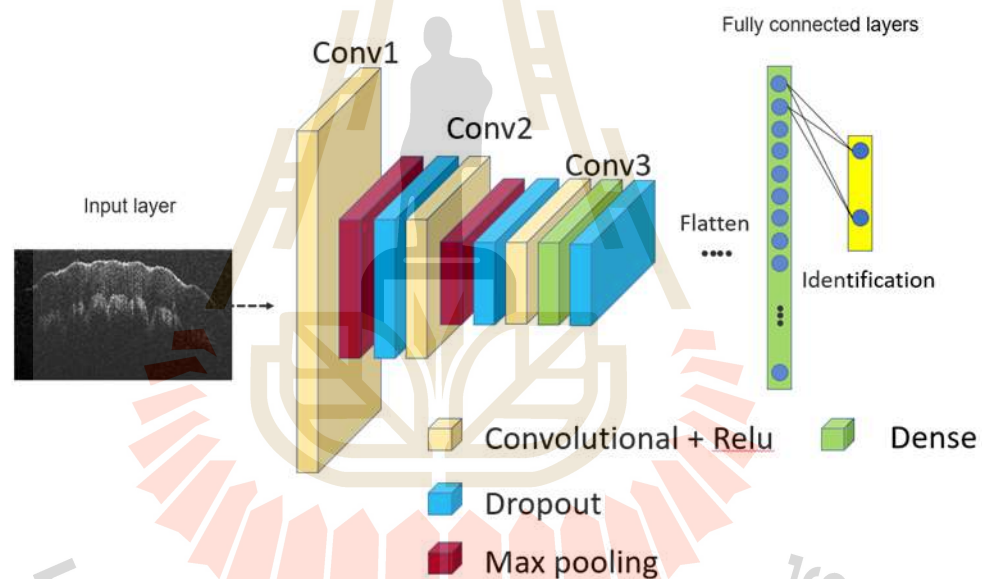


Figure 23 Neural network architecture diagrams of a model.

3.3 Training and validation procedures

The models were trained and validated using a consistent procedure, regardless of whether they were developed individually or via transfer learning. The dataset, defined as 1000 images, was obtained randomly from three imaging sessions of each finger. The dataset was divided into a training set containing 60% of the data and a validation set containing 40% of the data. During training, the weights of the models were modified using

backpropagation to minimize the categorical cross-entropy loss function. The models were compiled with the Adam optimizer, a batch size of 64, and 500 training epochs. The learning rate was set to decrease from 0.1 to 0.00001 by a factor of 10 after each training cycle to improve convergence. However, it was not possible to lower the learning rate further due to out-of-memory (OOM) errors. Similarly, the batch size could not exceed 64 due to OOM limitations. Additionally, the size of the input image was restricted to 128x128x3 due to OOM errors. It was not possible to reduce the size further as it could lead to the loss of important features and affect the model's performance. Furthermore, the use of pre-trained models constrained us to start with a size of 128x128x3, and going larger than this size was not possible due to OOM limitations. The model's performance was evaluated on the validation set using metrics such as accuracy, precision, recall, and F1-score. To merge the individual models, a voting process was used to determine the final prediction based on the prediction that received the most votes. The performance of the ensemble model was assessed on the validation set using the same metrics described in the literature review (2.3.3 Ensemble model). All the parameters are listed in Table 2.

Table 2 The parameter settings for four TL networks and a custom network.

| Networks | Parameters | | | | |
|--------------|--------------------|-----------------------|------------|-------|-----------|
| | Initial input size | Initial learning rate | Batch-size | Epoch | Optimizer |
| Inception-V3 | 128x128x3 | 0.00001 | 64 | 500 | Adam |
| Xception | 128x128x3 | 0.00001 | 64 | 500 | Adam |
| VGG-16 | 128x128x3 | 0.00001 | 64 | 500 | Adam |
| Restnet-50 | 128x128x3 | 0.00001 | 64 | 500 | Adam |
| Custom model | 128x128x3 | 0.00001 | 64 | 500 | Adam |

3.4 Model Evaluation and Comparison

After training and validation, the performance of the models was evaluated and compared using various metrics from the review in Chapter II. The individual models developed and the pre-trained transfer learning models were assessed for their accuracy, precision, recall, and F1-score. The evaluation aimed to determine the strengths and weaknesses of each model without disclosing specific results at this stage of the thesis. The analysis considered factors such as the effectiveness of transfer learning, the impact of different architectures, and the performance of ensemble methods. These findings will be presented and discussed in detail in the subsequent chapters dedicated to model evaluation and comparison, providing valuable insights into the performance and suitability of each model for the fingerprint recognition task.

3.5 Experimental Setup and hardware specifications

In this study, a portable, low-cost, and compact OCT imaging prototype was designed and implemented for use in field operations. The imaging system employed a light source with a central wavelength of 840 nm generated by a superluminescent light-emitting diode (SLED) from Box Optronics Technology Co., Ltd., China. The interferometer was based on the configuration of the free-space Michelson interferometer, with the interfered light beam coupled into a high-speed spectrometer for detection. A galvanometer mirror was used to synchronously scan the focused beam and obtain cross-sectional images and 3D datasets. Each cross-section had a resolution of 1000 by 900 pixels and 900 by 400 pixels, a scan area of approximately 4 mm by 4 mm, and a sampling distance of 4 μm between each lateral pixel. The system's depth resolution limit was approximately 10 microns based on the principle of low-coherence interferometry. The imaging speed was 25 frames per second. Overall, the OCT imaging prototype was designed with practicality and efficiency in mind, making it a suitable tool for fingerprint classification tasks in field operations.

3.5.1 Raw image size captured

This section presents the methodology employed to investigate the potential influence of image size on the accuracy of Optical Coherence Tomography (OCT) machine scans. The aim was to explore whether varying image sizes, which directly impact scanning speed, have any discernible effect on the accuracy of OCT-based diagnoses. This study employed a dataset of OCT images with different sizes, and several evaluation metrics were utilized to assess the performance of the machine learning models. To investigate the impact of image size on accuracy, a comprehensive dataset of OCT images was collected. The dataset included images of varying sizes, obtained by manipulating the dimensions of the original scans. Four specific image sizes were selected: 1000x600, 921x511, 409x511, and 150x510. These sizes were chosen to cover a wide range of dimensions commonly encountered in OCT machine scans. The experimental setup involved training a custom model using a dataset of OCT images with varying sizes. Custom model were employed to enable the model to learn intricate patterns and features in the OCT images. To assess the impact of image size on accuracy, the model was trained on images of different dimensions. The performance of the model was then evaluated using standard evaluation metrics, such as accuracy, precision, recall, and F1-score.

CHAPTER IV

RESULT AND DISCUSSION

Chapter IV presents the results and discussion of the experiments conducted to evaluate the performance of the proposed deep learning model for fingerprint classification using OCT images. This chapter describes the experimental setup, dataset, and evaluation criteria used to measure the performance of the model. The results are given and discussed with an emphasis on the model's capacity to identify various unique fingerprints. In addition, the approach's limitations and difficulties are examined, along with suggested areas for enhancement. The results reported in this chapter provide useful insights into the efficacy of utilizing OCT images with deep learning for fingerprint categorization.

4.1 Performance Evaluation Model

In this section, the performance of the proposed voting ensemble model for fingerprint classification using optical coherence tomography (OCT) images is evaluated. The model comprises five separate models: a Resnet50 pre-trained model, an Xception pre-trained model, a VGG16 pre-trained model, an InceptionV3 pre-trained model, and a custom model developed by the research team. The primary performance metric for the evaluation is the area under the curve (AUC) value on the receiver operating characteristic (ROC) curves, conducted on both a validation set and a holdout test set. Additionally, confusion matrices are analyzed to understand the model's ability to distinguish between multiple classes of internal fingerprints. The evaluation results provide a comprehensive assessment of the performance of the proposed model and demonstrate its potential usefulness in practical scenarios.

4.1.1 Learning curve

To evaluate the performance of the proposed voting ensemble model, five different models were trained on a dataset of 12,108 internal fingertip images captured using OCT scans. The dataset was randomly split into a 60% training set and a 40% validation set for each model, as discussed in the transfer learning section in Chapter II. The learning curves of each model are depicted below, which display the accuracy and loss values of each model as a function of the number of training epochs. The accuracy value indicates the percentage of correctly classified images, while the loss value represents the difference between the predicted and actual classification labels.

The learning curve offers valuable insights into the training process, as it helps determine whether the model is overfitting or underfitting the data. By analyzing the learning curve, the optimal number of epochs required for training the model to achieve the best performance can be identified. In the following sections, the results of each model are presented based on the learning curves.

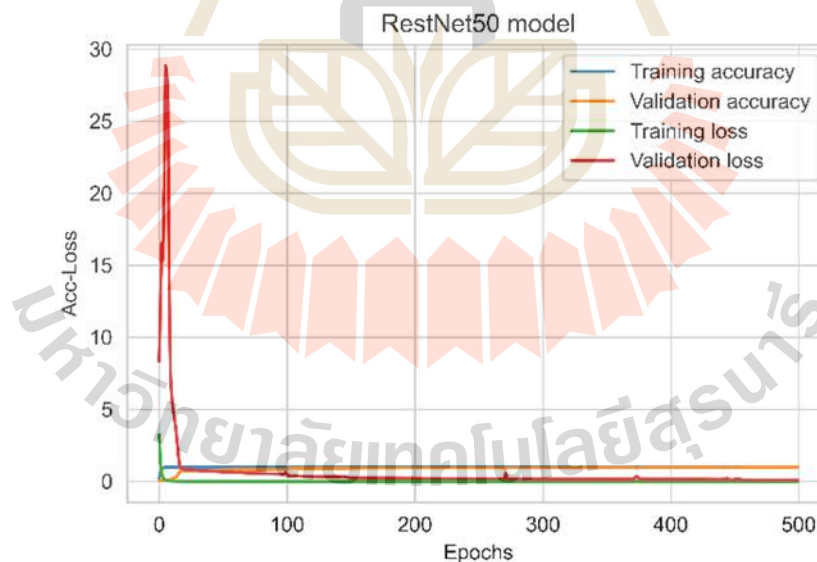


Figure 24 Learning curve of RestNet50 pre-trained model.

Figure 24 shows the acc-loss curve of the ResNet50 pre-trained model, which began with a high training and validation loss and a low training and validation accuracy.

However, the model quickly adjusted to the new data and, at epoch 8, had an accuracy of 0.99 and a loss of 0.034. After that, the training and validation loss spiked sharply, indicating that the model may have started to overfit. Nevertheless, the model completed the training phase with a high accuracy of 1. The validation accuracy kept a high level of consistency with a final value of 0.97 and a validation loss of 0.10. Despite some overfitting indicators, the ResNet50 model did a good job overall of accurately classifying the fingerprint images.

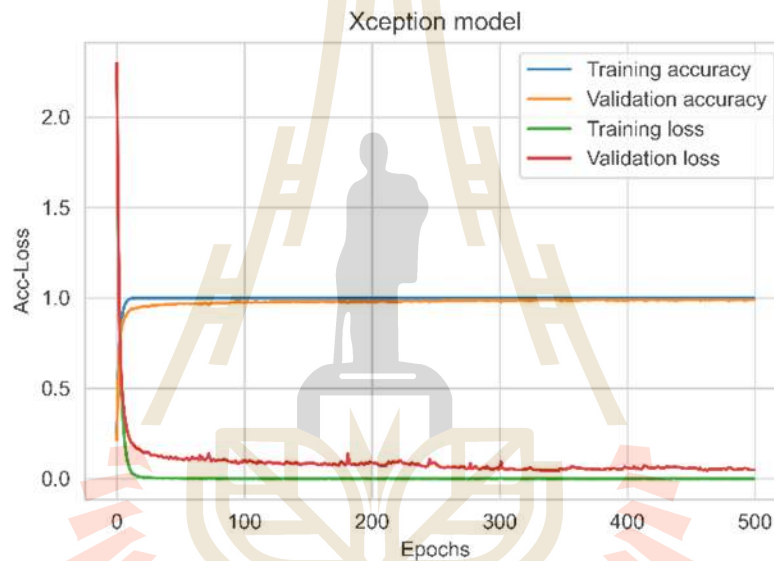


Figure 25 Learning curve of Xception pre-trained model.

Figure 25 shows the acc-loss curve of the Xception pre-trained model. The Xception model's learning curve has a slow start, low precision, and substantial loss at the beginning. But as the number of epochs rises, the model converges, and the accuracy of the training and validation data considerably increases as the loss falls. The model performs effectively and generalizes to unknown data when it achieves high accuracy on both the training and validation set with a very low loss after training.

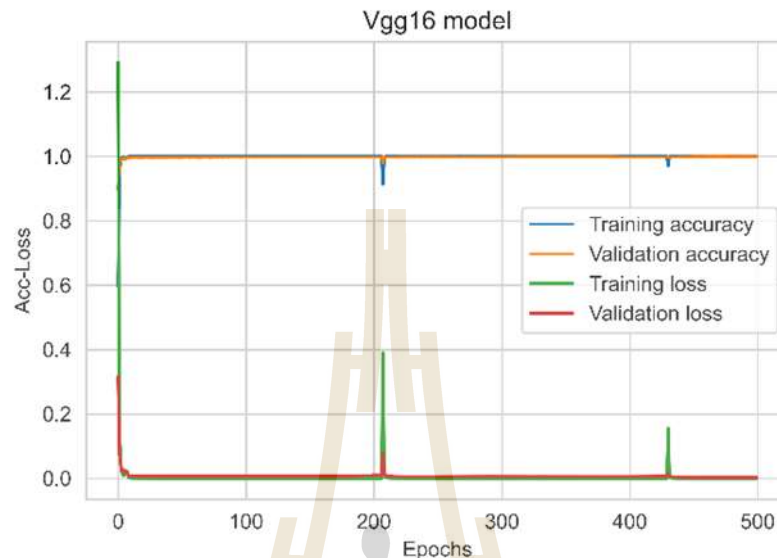


Figure 26 Learning curve of VGG16 pre-trained model.

Figure 26 shows the acc-loss curve of the *Xception* pre-trained model, From the initial epochs, both the training and validation accuracy grow dramatically while the loss declines. At epoch 207, the model reaches its highest level of precision with a training precision of 0.91, a loss of 0.39, a validation precision of 0.98, and a validation loss of 0.07. After the peak, the accuracy does not increase appreciably, however, the validation loss grows substantially at later epochs, indicating overfitting. The model achieved great accuracy on both the training and validation sets, with a final validation accuracy of 0.99 and a final training and validation loss of 0.002.

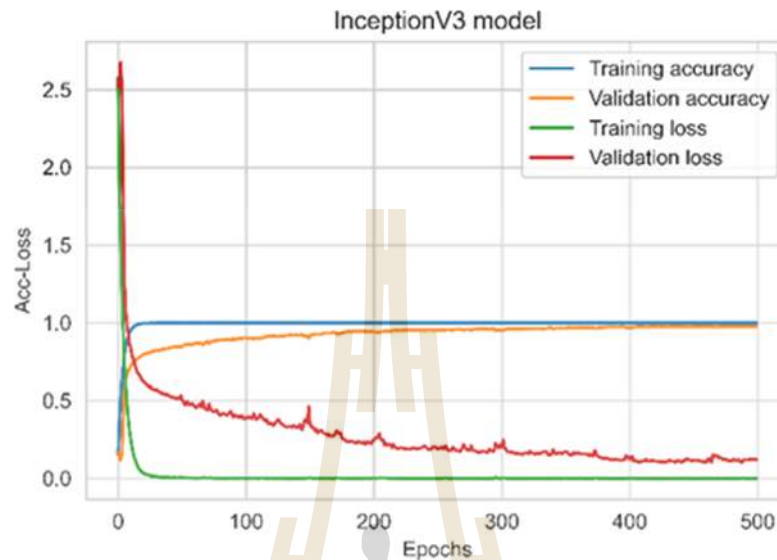


Figure 27 Learning curve of InceptionV3 pre-trained model.

Figure 27 displays the accuracy-loss curve of InceptionV3 during training. It was observed that both the accuracy and validation accuracy had a higher starting value. The validation accuracy of this model remained mostly stable, with only a minor decrease towards the end. The training and validation loss initially started high and decreased as the epoch values increased until they approached zero. In this model, however, the final training loss was smaller, suggesting a better capability to fit the data. Overall, the learning curve of the second model demonstrated its ability to handle the data effectively and achieve high accuracy with minimal loss values.

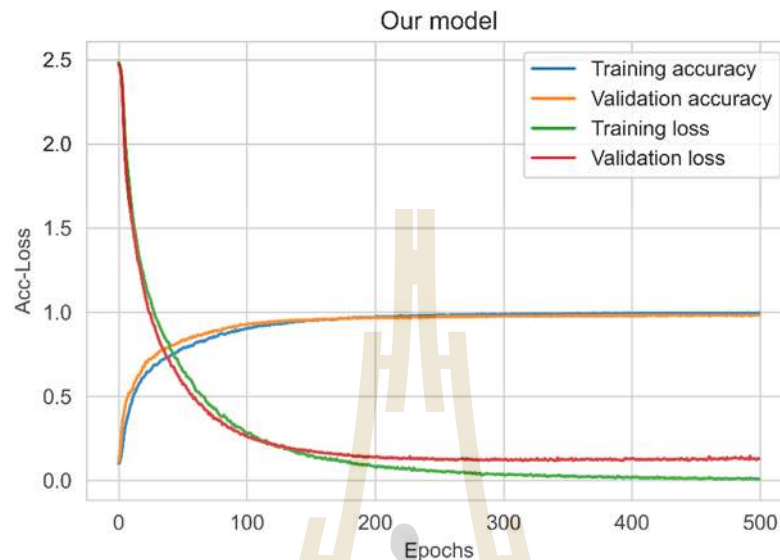


Figure 28 Learning curve of custom model.

Figure 28 illustrated the learning curve of the custom model, accuracy during training and validation both began at low levels and rose as the number of epochs increased. On the other hand, when the number of epochs rose, the training and validation loss began to decline. The model's weights' unpredictable initialization may have contributed to the initial significant loss. Weights were optimized while the model continued to train, which reduced loss. Along with the improvement in training accuracy, the validation accuracy also increased, showing that the model was not overfitting the training set of data. The fact that the validation data was not used for weight optimization may also be the reason why the validation loss began out higher than the training loss and stayed somewhat higher throughout training.

4.1.2 Confusion matrix

A confusion matrix is a vital tool for evaluating the performance of a classification model. It displays the number of correct and incorrect predictions made by the model, providing insights into its effectiveness. The matrix includes four categories: true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN).

Table 3 shows the confusion matrix for ResNet50, the ResNet50 model generally delivered accurate predictions, which suggests a good level of accuracy overall. Users 1, 3, 5, and 10 are tiny-problematic for the model since each of these classes has a significant amount of incorrect predictions. The model is most accurate for users 4, 7, and 8, with all predictions being correct.

Table 4 Table of confusion matrix table for Xception model.

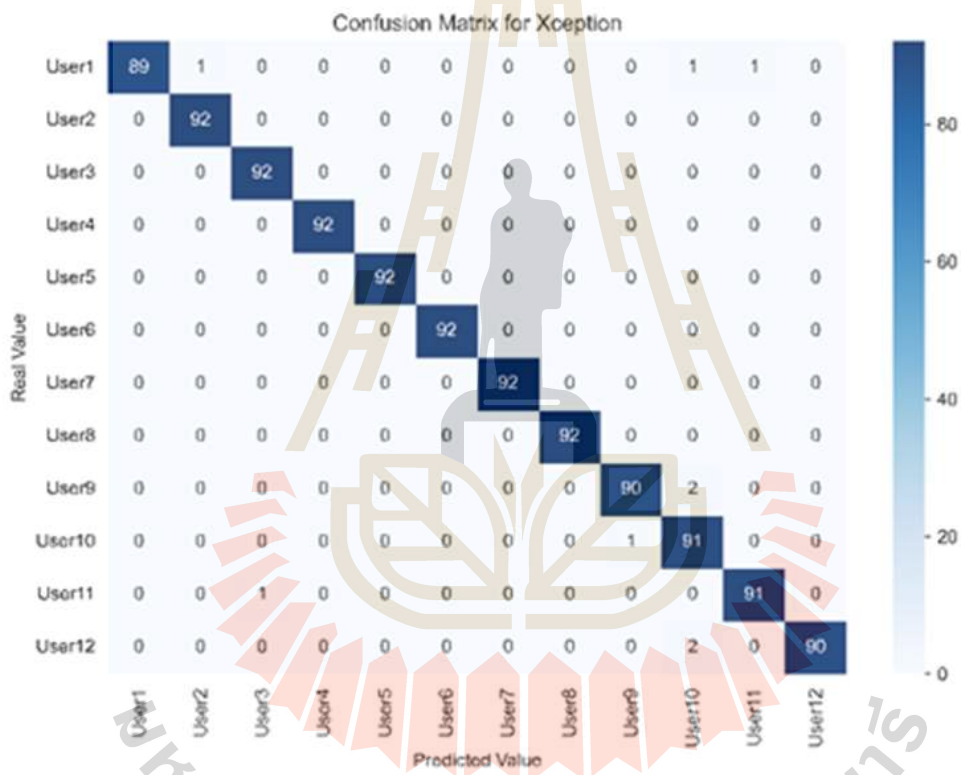


Table 4 shows the confusion matrix for the Xception model, the Xception model is the most accurate overall and can correctly forecast the majority of courses. There are a few inaccuracies in the projections for users 1, 9, 10, 11, and 12, where the model incorrectly identified one or two data sets as belonging to a different user. The model's accuracy is highest for user 2, 3, 4, 5, 6, 7, and 8; all predictions were realized.

Table 5 Table of confusion matrix table for VGG16 model.

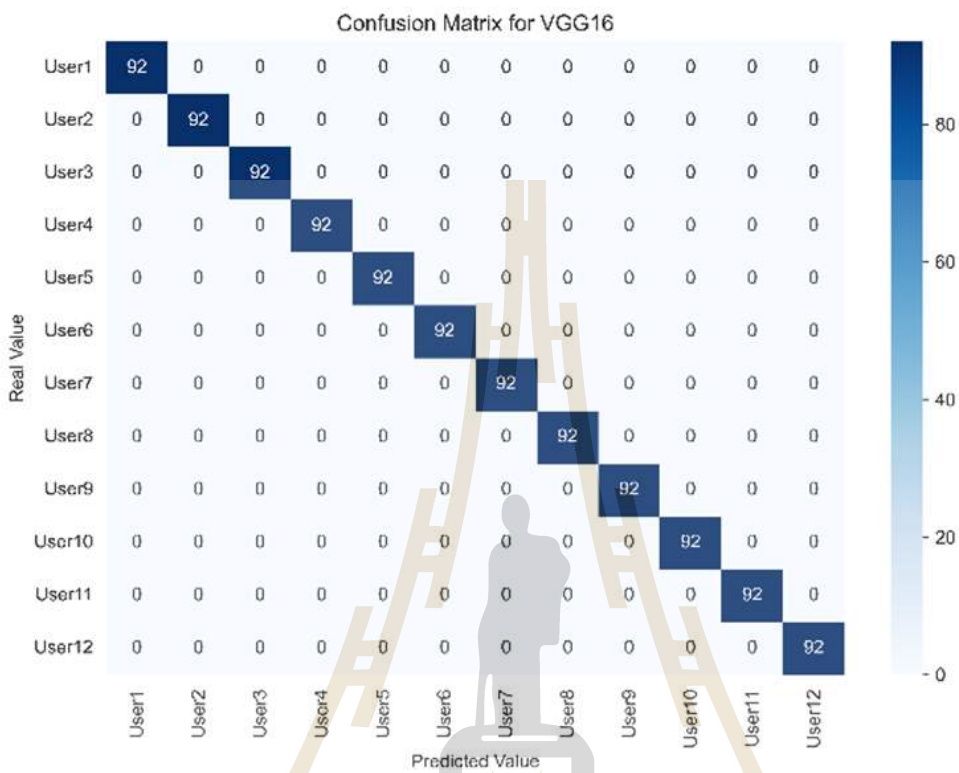


Table 5 visualized the confusion matrix of Vgg16, with the majority of forecasts being true, the VGG16 model has good overall accuracy. Users 1, 5, and 11 are problematic for the model, as each of these classes has a significant amount of incorrect predictions. With every prediction coming true for user 4, the model has the best accuracy.



Table 6 Table of confusion matrix table for InceptionV3 model.

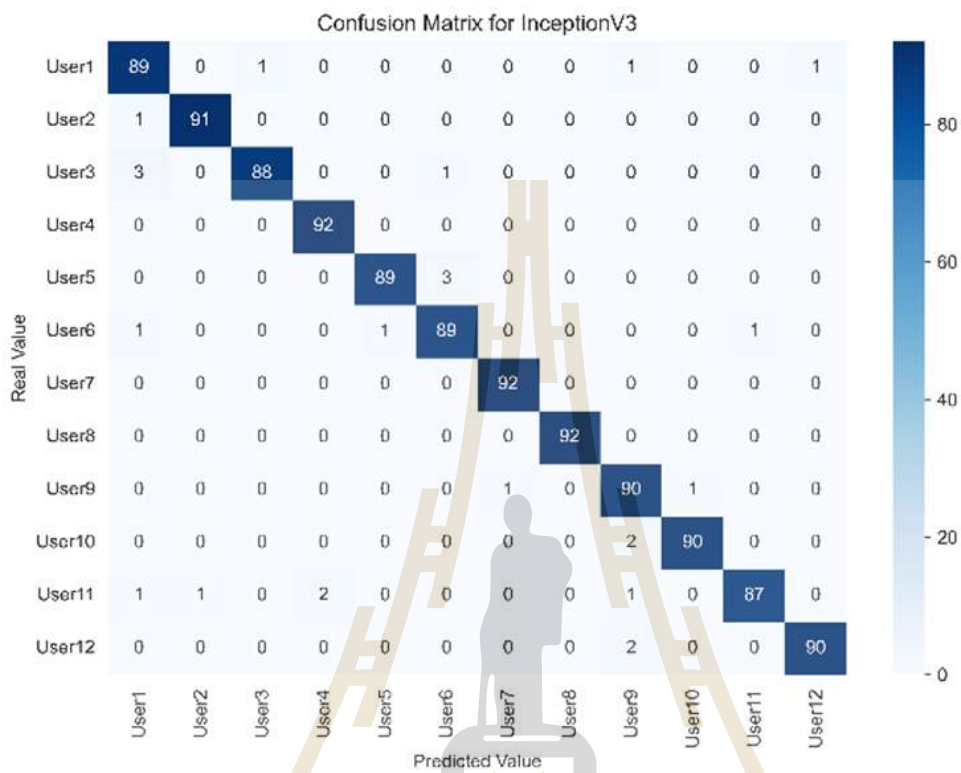


Table 6 illustrated the confusion matrix for InceptionV3, The InceptionV3 model achieved accurate predictions, showing excellent overall accuracy. Users 1, 3, 5, and 11 are problematic for the model, as each has some erroneous predictions. Users 4 and 7 had the most accurate forecasts, with all predictions being accurate.

Table 7 Table of confusion matrix table for custom model.

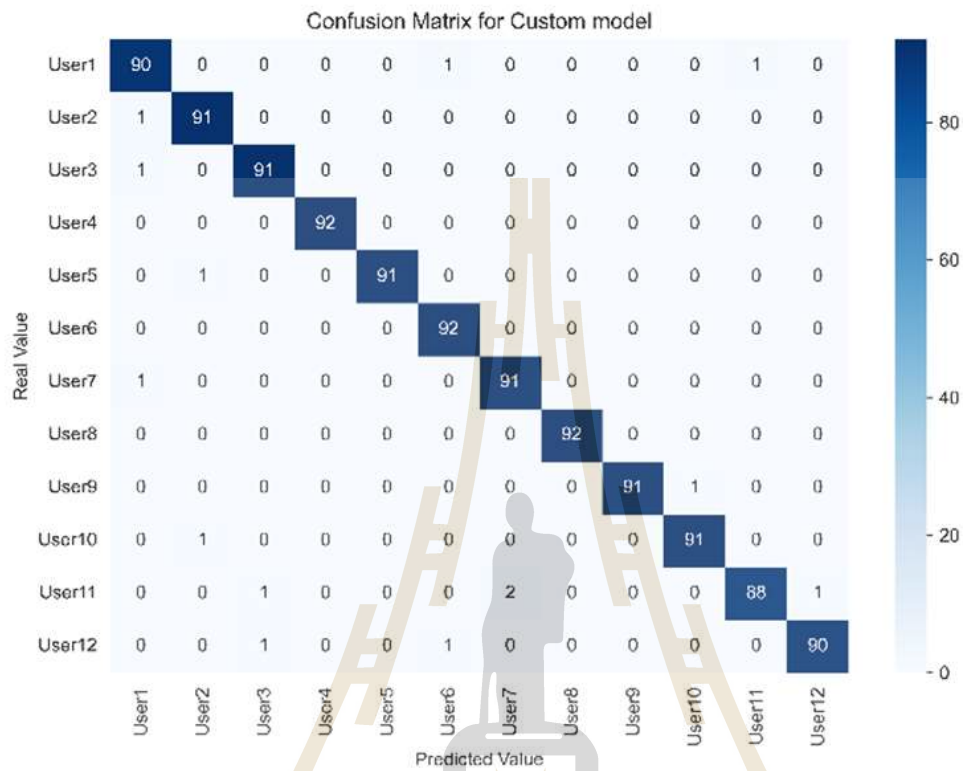


Table 7 shows the confusion matrix of the custom model, the overall accuracy of the custom model is demonstrated by the significant number of accurate predictions (at least 90) for the majority of classes. There are errors in the predictions, especially for users 1, 6, and 11, for whom the model misclassified one or two observations. For user 4, the model's accuracy is the highest, since all predictions are accurate.

4.2 Model comparison

Let's consider the learning curve, the curves of the five models show a similar pattern of convergence in which training and validation losses drop with increasing epochs while training and validation accuracies improve. However, the particular contours of the curves varied amongst models, presumably due to variances in their designs. The ResNet50

and Xception models begin with relatively low precision and significant losses but ultimately attain high precision and low losses.

Considering the confusion matrix, it presents a summary of the performance of each model. All five models were able to predict with an accuracy of over 97 percent. The data from the confusion matrix can be utilized to calculate the classification report for comparing the models. The calculation of the classification report was reviewed in Chapter II (Section 2.3.4 Performance Evaluation). The table below displays the classification report of the five models, providing the data for accuracy, precision, recall, and F1 score, which are used to assess a classifier's performance.

Table 8 Comparison of Classification report between five models and voting method.

| Network | Accuracy | Precision | Recall | F1-score | Support |
|---------------|----------|-----------|--------|----------|---------|
| Inception-V3 | 97.59 | 97.65 | 97.61 | 97.62 | 1104 |
| Xception | 99.13 | 99.2 | 99.18 | 99.19 | 1104 |
| VGG-16 | 100 | 100 | 100 | 100 | 1104 |
| Resnet-50 | 97.1 | 97.18 | 97.1 | 97.12 | 1104 |
| Custom model | 98.65 | 98.66 | 98.69 | 98.67 | 1104 |
| Voting method | 100 | 100 | 100 | 100 | 1104 |

This table presents a comparison of the performance of five different models and a voting method used to identify users based on OCT scans of their internal fingertips. The accuracy of the models ranged from 97.1% to 100%, with the VGG-16 model achieving perfect accuracy. Precision, recall, and F1-score were all above 97%, with the Xception model achieving the highest values for these metrics. The custom model achieved an accuracy of 98.65%, which is slightly lower than the Xception model's accuracy of 99.13%. However, the voting method achieved perfect accuracy, indicating that it was able to leverage the strengths of each model to correctly identify every user in the dataset. This

table provides important insights into the strengths and weaknesses of each model and highlights the effectiveness of using an ensemble method to improve the accuracy of the classification system.

With the aim of maximizing accuracy, the voting method was chosen as a rational approach in this research. All five models exhibited high accuracy rates, making the voting method an appropriate choice to combine their predictions. Table 8 presents the confusion matrix, showcasing the performance of our system. Notably, the voting method outperformed each model, achieving the highest accuracy. Remarkably, our system achieved a perfect 100% accuracy, correctly identifying the user for every fingerprint impression. These exceptional results highlight the effectiveness and robustness of our proposed system in accurately identifying users based on internal fingertip images. The significant implications of such high accuracy rates extend to various applications, particularly in the domains of application with security and access control systems.

4.3 Analysis of Receiver Operating Characteristic (ROC) Curves

In addition to using confusion matrices to evaluate the performance of a classification model, another widely used method is to analyze the Receiver Operating Characteristic (ROC) curve. The ROC curve is a graphical representation of a binary classifier's performance that shows the tradeoff between the true positive rate (TPR) and the false positive rate (FPR) for different classification thresholds. The ROC curve is an important tool for assessing the performance of a classification model across a range of thresholds and can help identify the optimal threshold for the model. In this section, the ROC curves of the different models used in this study are analyzed to determine their performance and identify the optimal threshold for each model.

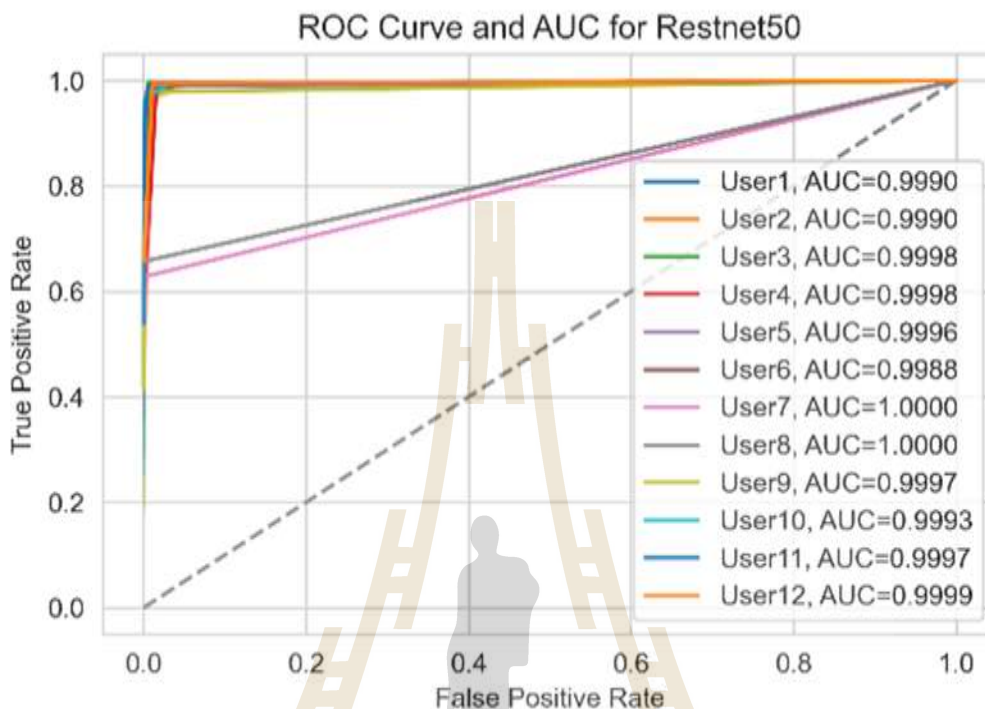


Figure 29 ROC and AUC can visualize the performance of ResNet50.

Figure 29 illustrates ResNet50 performance: The AUC of this model, which is also fairly good at 0.9997, indicates that it distinguishes between positive and negative classifications. The ROC curve illustrates that while the FPR increases, the TPR increases rapidly, suggesting the model's high sensitivity and capacity to recognize true positives with a comparatively low rate of false positives. When the TPR reaches 1, the curve flattens, indicating that the model's classification performance at that level is optimal.

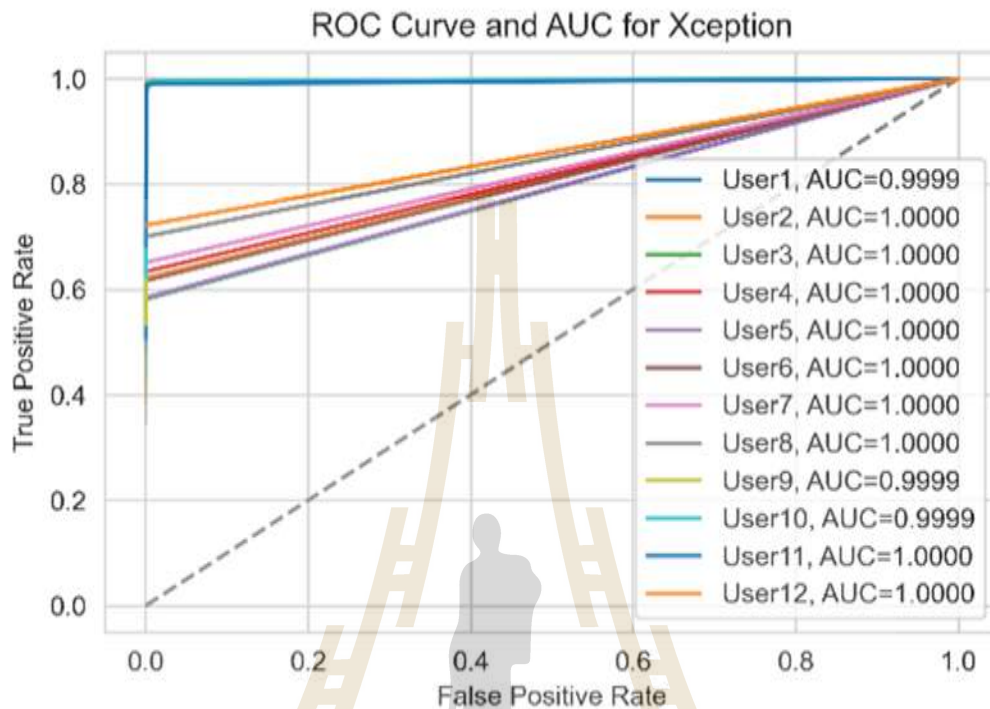


Figure 30 ROC and AUC curve visualize the performance of Xception model.

Figure 30 depicts Xception's performance: This model's AUC is 1.0, the maximum possible value, exhibiting a remarkable ability in distinguishing between positive and negative classifications. The ROC curve illustrates the model's capacity to achieve optimum classification performance, with a TPR of 1 and an FPR of nearly 0, showing that the model has a high degree of sensitivity and specificity.

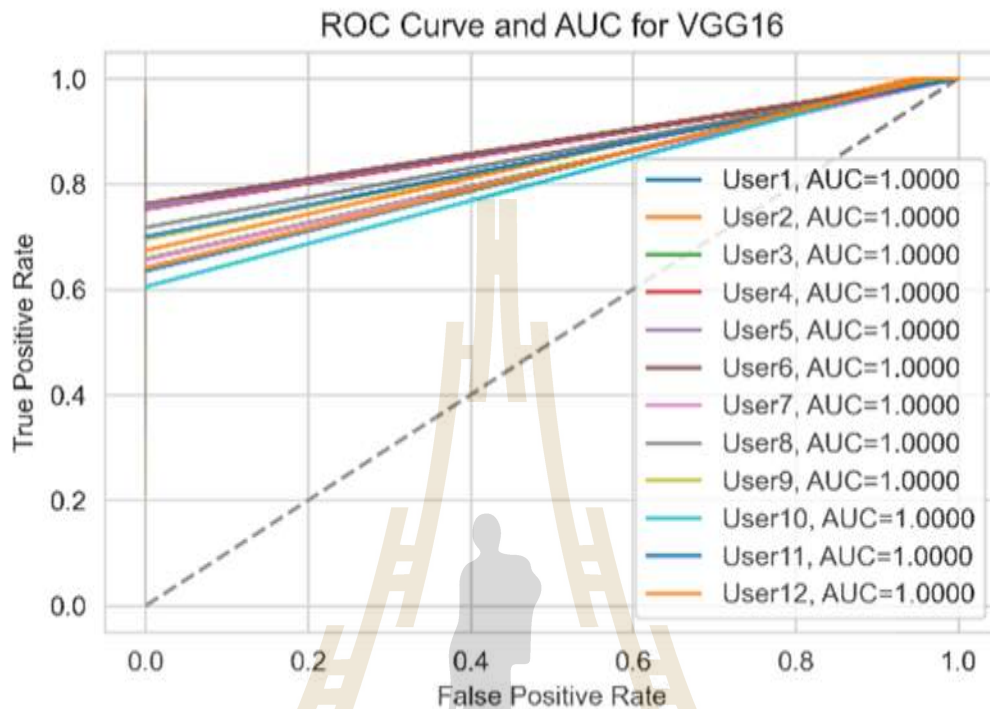


Figure 31 ROC and AUC curve visualize the performance of VGG16 model.

Figure 31 illustrates the effectiveness of VGG16: This model distinguishes between positive and negative classes with an AUC of 1, the maximum attainable value, exhibiting outstanding performance. The ROC curve illustrates the model's capacity to achieve optimum classification performance, with a TPR of 1 and an FPR of nearly 0, showing that the model has a high degree of sensitivity and specificity.

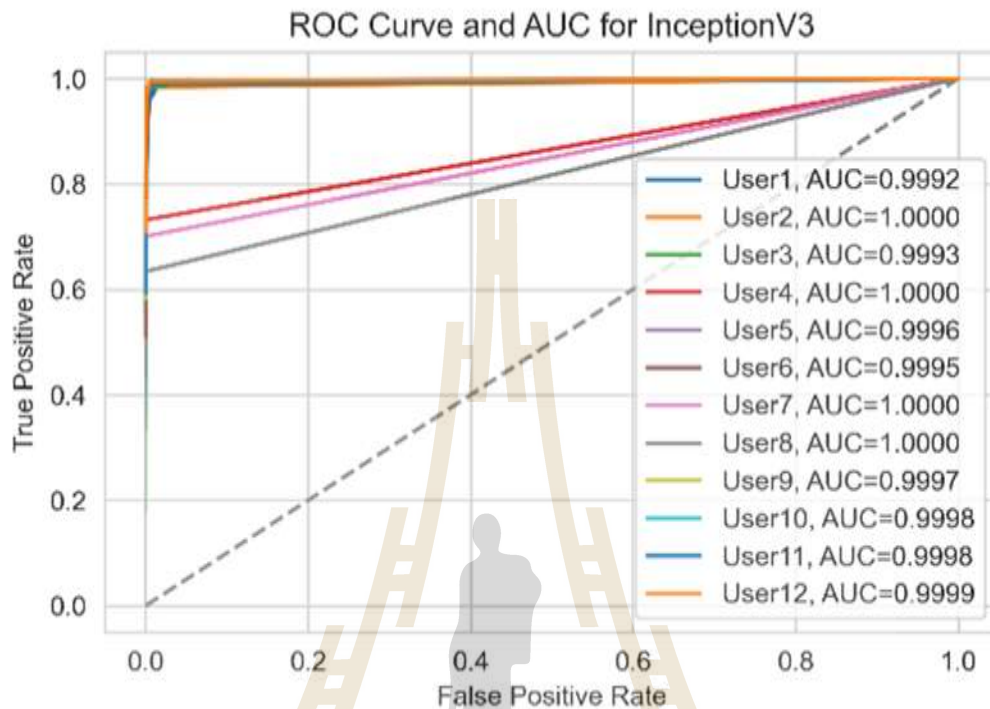


Figure 32 ROC and AUC curve visualize the performance of InceptionV3.

Figure 32 exhibits InceptionV3's effectiveness: The AUC of this model is 0.9998, which is more than the AUC of the custom model, indicating that it performs extraordinarily well at differentiating positive and negative classes. The ROC curve illustrates that while the FPR grows, the TPR climbs rapidly, confirming the model's high sensitivity and ability to recognize true positives with a low rate of false positives. When the TPR reaches 1, the curve flattens, indicating that the model's classification performance at that level is optimal.

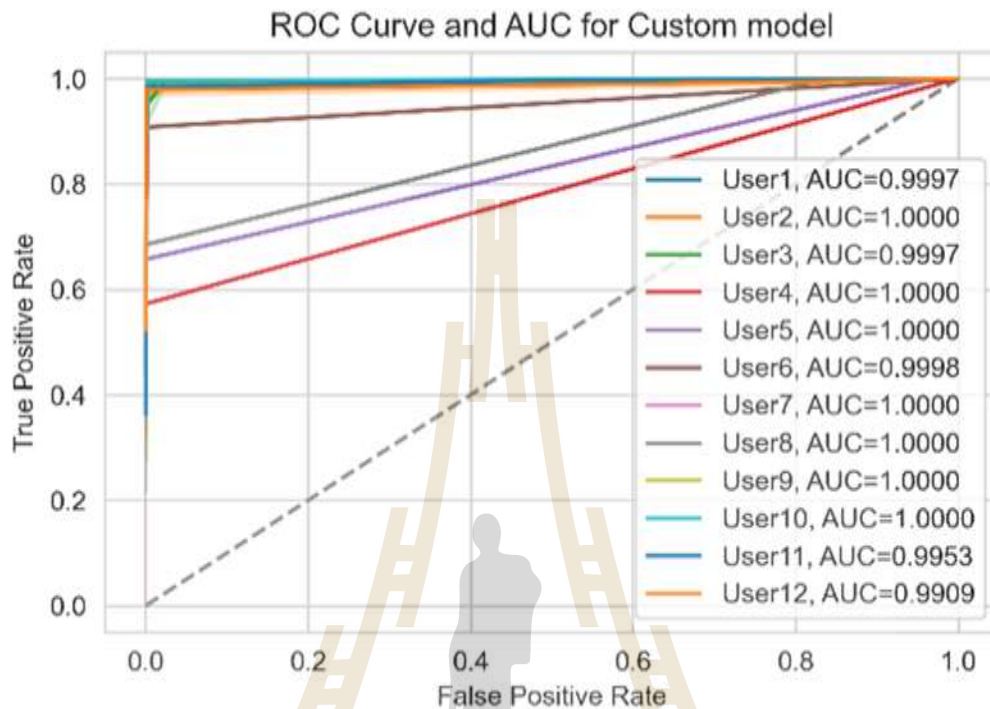


Figure 33 ROC and AUC curve visualize the performance of the custom model.

Figure 33 shows the individually designed model: The AUC for this model is 0.9953, showing that it distinguishes between positive and negative classifications adequately. The ROC curve illustrates that while the FPR grows, the TPR climbs rapidly, demonstrating that the model has a high sensitivity and can recognize true positives with a low rate of false positives. When the TPR reaches 1, the curve flattens, indicating that the model's classification performance at that level is optimal.

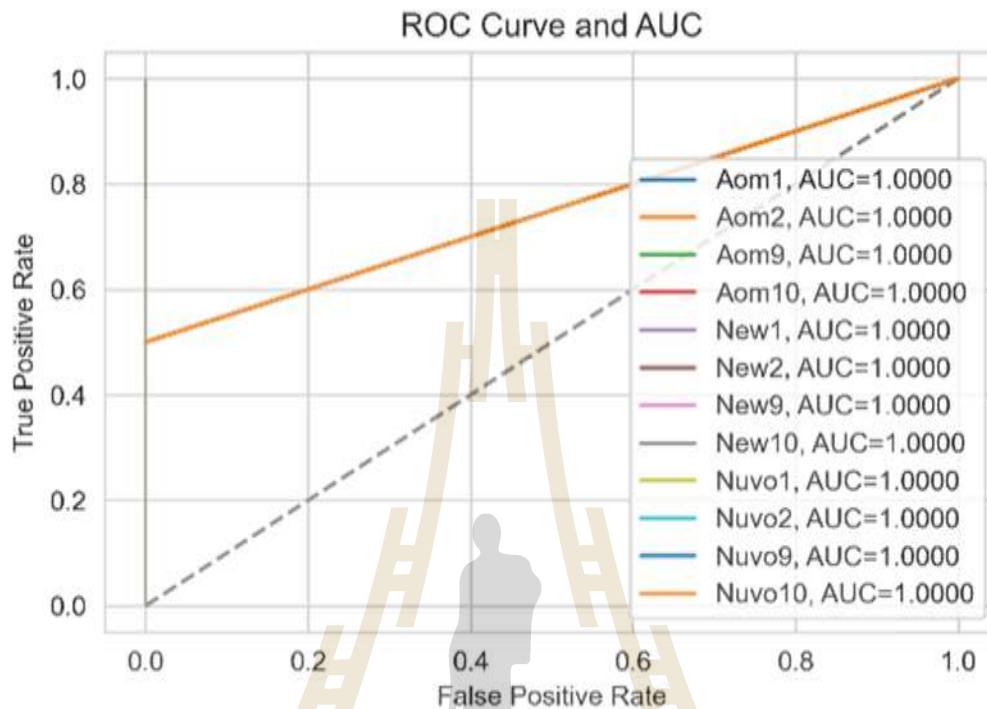


Figure 34 ROC and AUC curve visualize the performance of Ensemble model.

Figure 34 shows the ROC and AUC curve of the ensemble model (voting method), the result of the ensemble voting method showed that the ROC curves of the 12 classes overlapped in the same position, indicating that the model was able to effectively distinguish between each class with high accuracy. Furthermore, the AUC for each class was 1, indicating that the model achieved perfect performance for each class. These results suggest that the proposed ensemble voting model was able to effectively leverage the strengths of each model to achieve superior classification performance.


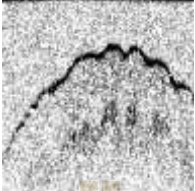
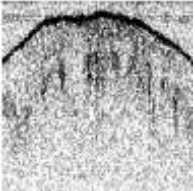
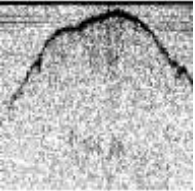

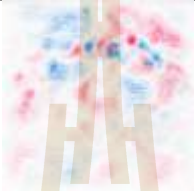


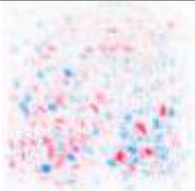

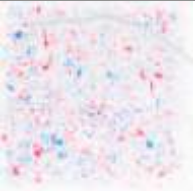













The analysis of ROC curves offered valuable insights into the performance of the proposed ensemble voting model. High AUC scores for each class indicated that the model accurately classified each sample, while the consistent overlap of the ROC curves suggested that the model's performance was maintained across all classes. These findings demonstrate that the proposed ensemble voting model is a reliable and efficient approach for classifying internal images of fingertips using OCT technology.

4.4 Explainable model

In this chapter, the results of the project, "Identifying Human Fingerprints Using Optical Coherence Tomography (OCT) Images with Deep Learning," are presented. The primary objective of this study is to develop a robust and accurate fingerprint identification system that utilizes deep learning models to analyze OCT images. The use of OCT imaging provides a promising approach for fingerprint identification, as it captures high-resolution data that can effectively distinguish individual fingerprints. To ensure the credibility and interpretability of the models, explainable AI (XAI) techniques were incorporated, enabling an understanding of each model's decision-making process and facilitating performance assessment by SHAP, as detailed in the table below. Referring to Chapter II, we selected the SHAP instead of LIME and Grad-CAM due to the SHAP offers a versatile approach for producing reliable and understandable explanations for complex models.

In this study, the trustworthiness and interpretability of five distinct models were analyzed: 1) a custom model, 2) InceptionV3, 3) VGG16, 4) Xception, and 5) ResNet50. The SHAP method was employed for feature importance and model explanation. It provided valuable information about the decisions made by these models when presented with randomly selected, unseen data. The SHAP method generates explanations using a color-coded system, where each color represents the impact of a specific feature on the model's prediction. Positive values (red) indicate that the presence of the feature increases the prediction likelihood, while negative values (blue) suggest that the presence of the feature decreases the prediction likelihood. The magnitude of the color intensity corresponds to the strength of the impact on the prediction. The following table offers a comprehensive explanation of the feature selection process. To illustrate the differences in how each model interprets the input data, four sample images were randomly chosen from the unseen OCT fingerprint dataset.

Table 9 Explainable model table shows the feature selection by SHAP.

| | | | | |
|----------------|---|---|--|---|
| Original image |  |  |  |  |
| Custom model |  |  |  |  |
| InceptionV3 |  |  |  |  |
| ResNet50 |  |  |  |  |
| VGG16 |  |  |  |  |
| Xception |  |  |  |  |

The custom model placed a high emphasis on the surface features of the fingerprint, as indicated by the concentrated red zone on the fingertip surface area. In contrast, the InceptionV3 and ResNet50 models exhibited a broader distribution of red regions throughout the images, considering various parts of the image in addition to the fingerprint surface. The VGG16 model had a targeted focus on crucial features, including

the fingerprint surface, sweat glands, and sweat holes, allowing for highly informed predictions. The Xception model displayed a distribution of red clusters similar to InceptionV3 and ResNet50, indicating its consideration of multiple aspects of the image for identity prediction.

These findings highlight the varying interpretability of the models and how they prioritize different features during the decision-making process. Such insights provided by the SHAP method contribute to a better understanding of the models' reasoning and support their trustworthiness in real-world applications.

4.4 Image size effect

This section presents the results obtained from the evaluation of image size on the classification report of OCT machine scans. The performance of the machine learning model was assessed using various evaluation values for four different image sizes. The findings provide insights into the relationship between image size and accuracy in OCT-based diagnoses. The result will show in the table.

Table 10 Table of classification reports for different image sizes.

| Image size (Pixels) | Accuracy | Precision | Recall | F1-score |
|---------------------|----------|-----------|--------|----------|
| 1000*600 | 0.9947 | 0.9948 | 0.9947 | 0.9947 |
| 921*511 | 0.9634 | 0.964 | 0.9634 | 0.9632 |
| 409*511 | 0.9947 | 0.9948 | 0.9947 | 0.9947 |
| 150*510 | 0.9683 | 0.9697 | 0.9683 | 0.9681 |

Table 10 summarizes the performance metrics achieved by the machine learning model for each image size. The metrics include accuracy, precision, recall, and F1-score. Surprisingly, the size was reduced to a very small 150x510, the results demonstrate that there is no significant impact of image size on the accuracy of OCT machine scans.

Regardless of the image size, the model achieved consistently high performance across all metrics, with accuracy ranging from 0.9634 to 0.9947.

4.5 Applications

The findings from this study, which leverages three-dimensional OCT-based fingerprint recognition, demonstrate the potential for significantly enhancing security systems. While the results are promising, it is important to acknowledge the limitations, including the small sample size and the reliance on deep learning techniques. To effectively apply this work to real-world security systems, further research, and development are needed to design a robust and scalable system tailored to specific safety requirements. Future advancements should focus on addressing these limitations and exploring additional techniques to optimize the performance, scalability, and adaptability cross-sectional of three-dimensional OCT-based fingerprint recognition for a wide range of security applications.

Preliminary results indicate that the developed system based on the depth profile of three-dimensional OCT image analysis is capable of real-time person identification. The system effectively distinguishes individuals using OCT images captured by an active OCT camera system specifically designed for fingerprint recognition. This promising outcome demonstrates the potential feasibility and practicality of employing OCT technology for seamless and efficient fingerprint recognition in real-world security applications as visualized in Figure 39. However, further research and testing on larger datasets and under various environmental conditions are necessary to validate and optimize the system's performance, ensuring its reliability and robustness in diverse operational settings.

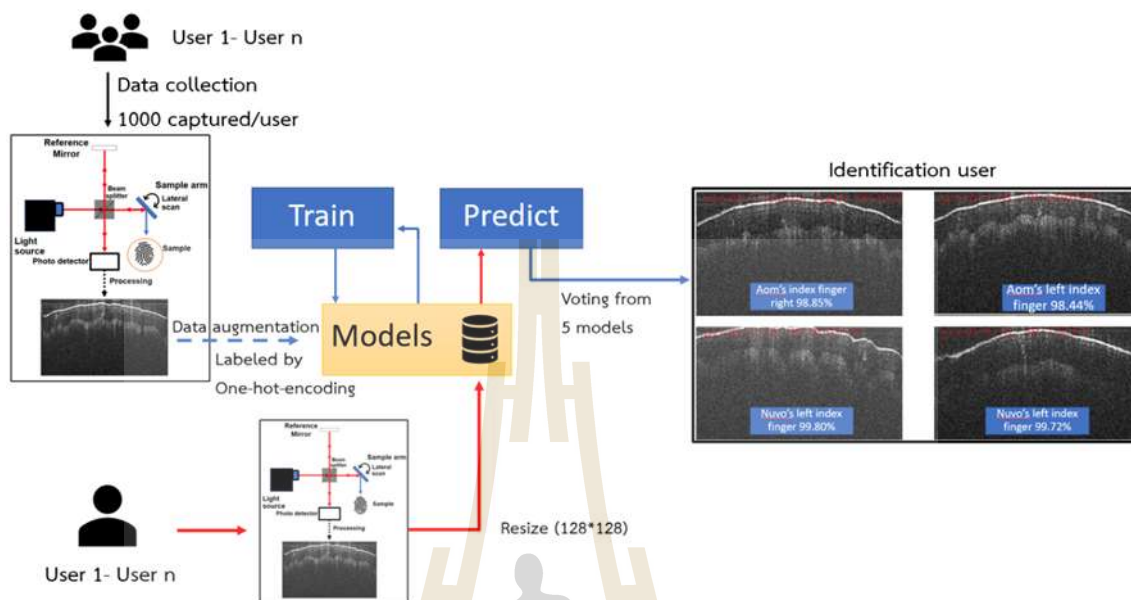


Figure 35 Example for the application workflow diagram.

4.6 Discussion of Limitations and Future Directions

This study has various limitations that must be taken into account. First, the dataset used in this study was acquired from a single institution, which may limit the applicability of the findings to other groups. In addition, the dataset was of limited size, which may have impacted the performance of the models. Additional research using larger and more diverse data sets is required to confirm the results of this study. Another limitation is the use of a single type of OCT machine. The models developed in this study were trained and tested only on OCT images obtained from one type of machine, which may limit their applicability to other OCT machines. Future studies should evaluate the generalizability of the developed models to other OCT machines.

Despite the promising results of our internal fingertip identification system using OCT, several limitations must be addressed. One limitation is the need for high-quality images to achieve accurate identification, which can be challenging to obtain in real-world scenarios due to factors such as finger positioning and movement. Additionally, our system was developed and tested on a dataset of 12 users, which may limit the generalizability of the results to a larger population.

Plans include expanding our dataset to include a more varied group of users and including additional parameters such as finger shape and texture to increase the recognition accuracy. The use of biometric data, such as fingerprints, for identification, increases privacy concerns, making the potential consequence of security breaches an additional crucial factor. As a result, additional research is required to build effective security mechanisms that safeguard user data while allowing for precise and rapid identification. Consequently, our internal fingertip identification system shows promise as a non-invasive and convenient biometric authentication approach; however, additional study and improvement are required to solve the aforementioned constraints and assure its practical applicability in real-world situations.

In the future, it may be worth exploring the possibility of tracking sweat glands for identification purposes, as wetness and dirt on the fingertip can impact the performance of the model. This could involve investigating new image processing techniques or incorporating additional sensor data to account for such environmental factors. These advancements would contribute to the practical applicability of our internal fingertip identification system, offering a non-invasive and convenient biometric authentication approach. However, it is essential to continue studying and refining the system to overcome the aforementioned limitations and ensure its effectiveness in real-world scenarios.

CHAPTER V

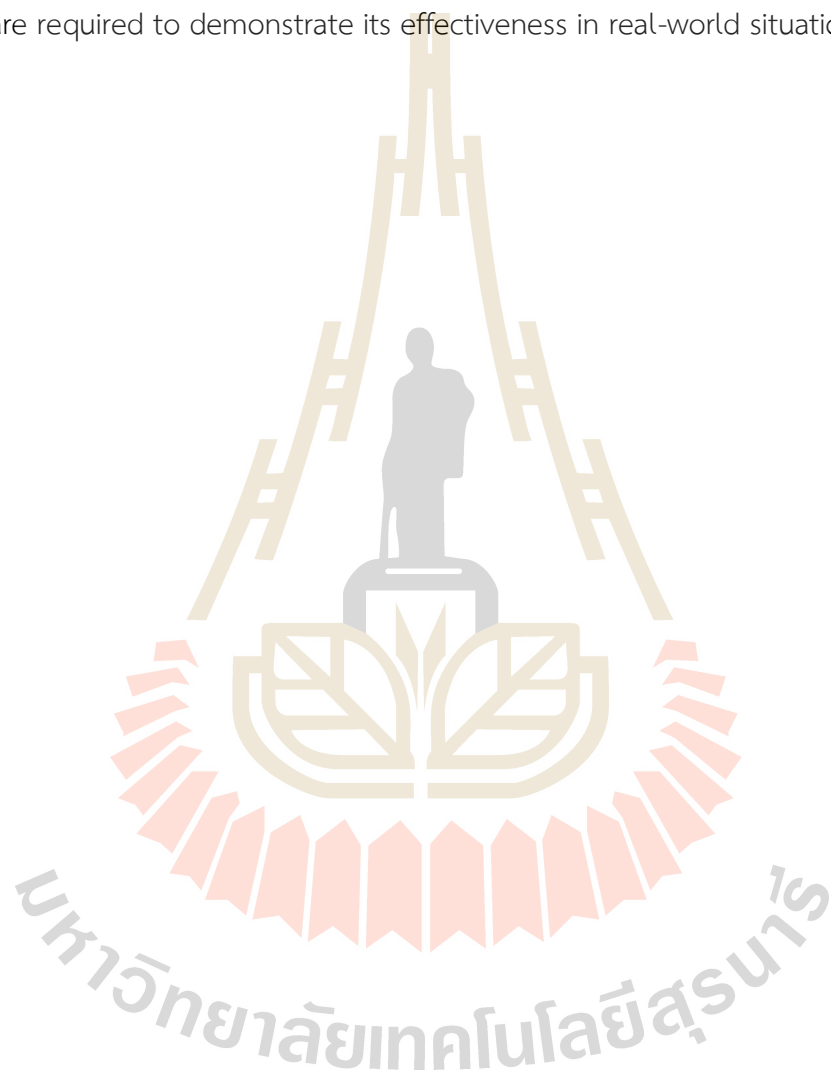
CONCLUSION

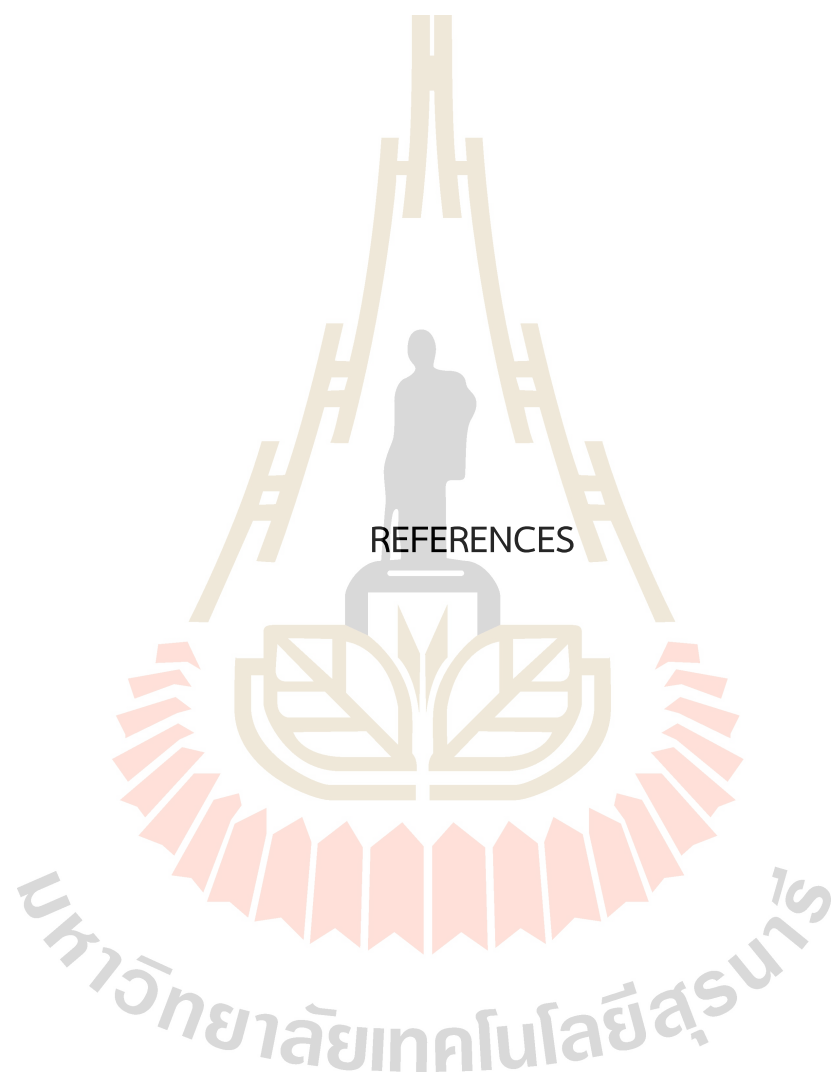
In this research, the use of optical coherence tomography (OCT) was explored for application for the performance and security of traditional fingerprint scanners in the field of finger biometrics. A security system was proposed that leverages the internal structure of the fingertip, including sweat glands, epidermis, and dermis, to create an internal image of the finger. Pre-trained deep learning models, such as Inceptionv3, VGG16, Xception, and ResNet50, along with a custom model, were utilized to extract features from the internal fingertip images. A voting method was employed to ensure the robustness of the system.

Experimental results demonstrated that the proposed method achieves an accuracy and prediction rate of around 99%, according to the confusion matrix evaluation. This evidence suggests that a security system based on deep learning with internal fingertip data has the potential to help traditional verification systems, offering higher security. However, some limitations should be considered, such as the relatively small sample size of 12 unique fingers. Expanding data collection and analysis with a larger sample size could enhance the accuracy and robustness of the proposed method. Moreover, the system was tested under controlled laboratory conditions; further testing under more diverse and challenging conditions is necessary to confirm its effectiveness in real-world scenarios.

Future studies could focus on applying the proposed technique to other biometrics, such as voice or facial recognition, and integrating the system into actual devices with internal OCT hardware. It would also be worthwhile to investigate the potential for improving system security by exploring various deep-learning approaches and model designs.

In conclusion, the proposed security solution, which combines internal fingertip data with deep learning, has the potential to significantly enhance the efficiency and apply to the security of conventional fingerprint scanners. Future research may explore new avenues for improving the system's accuracy and security, but additional validation and testing are required to demonstrate its effectiveness in real-world situations.





REFERENCES

- Adadi, A., and Berrada, M. (2018). Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). *IEEE Access*, 6, 52138-52160.
- Agarap, A. F. (2018). Deep Learning using Rectified Linear Units (ReLU).
- Akbari, N. (2012). Automation of Fingerprint Recognition Using OCT Fingerprint Images. *Journal of Signal and Information Processing*, 03, 117-121. doi:10.4236/jsip.2012.31015
- Almahdi, R., and Ragb, H. (2019). *Fused Deep Convolutional Neural Networks Based on Voting Approach for Efficient Object Classification*.
- Altmann, A., Tološi, L., Sander, O., and Lengauer, T. (2010). Permutation importance: a corrected feature importance measure. *Bioinformatics*, 26(10), 1340-1347.
- Barredo Arrieta, A., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., . . . Herrera, F. (2020). Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 58, 82-115. doi:https://doi.org/10.1016/j.inffus.2019.12.012
- Biggio, B., Akhtar, Z., Fumera, G., Marcialis, G. L., and Roli, F. (2012). Security evaluation of biometric authentication systems under real spoofing attacks. *IET Biometrics*, 1(1). doi:10.1049/iet-bmt.2011.0012
- Bouma, B. E., and Tearney, G. J. (2002). *Handbook of optical coherence tomography*: New York : Marcel Dekker.
- Chollet, F. (2017). Xception: Deep Learning with Depthwise Separable Convolutions. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1800-1807.
- Chugh, T., and Jain, A. K. (2019a). Fingerprint Presentation Attack Detection: Generalization and Efficiency. *2019 International Conference on Biometrics (ICB)*, 1-8.

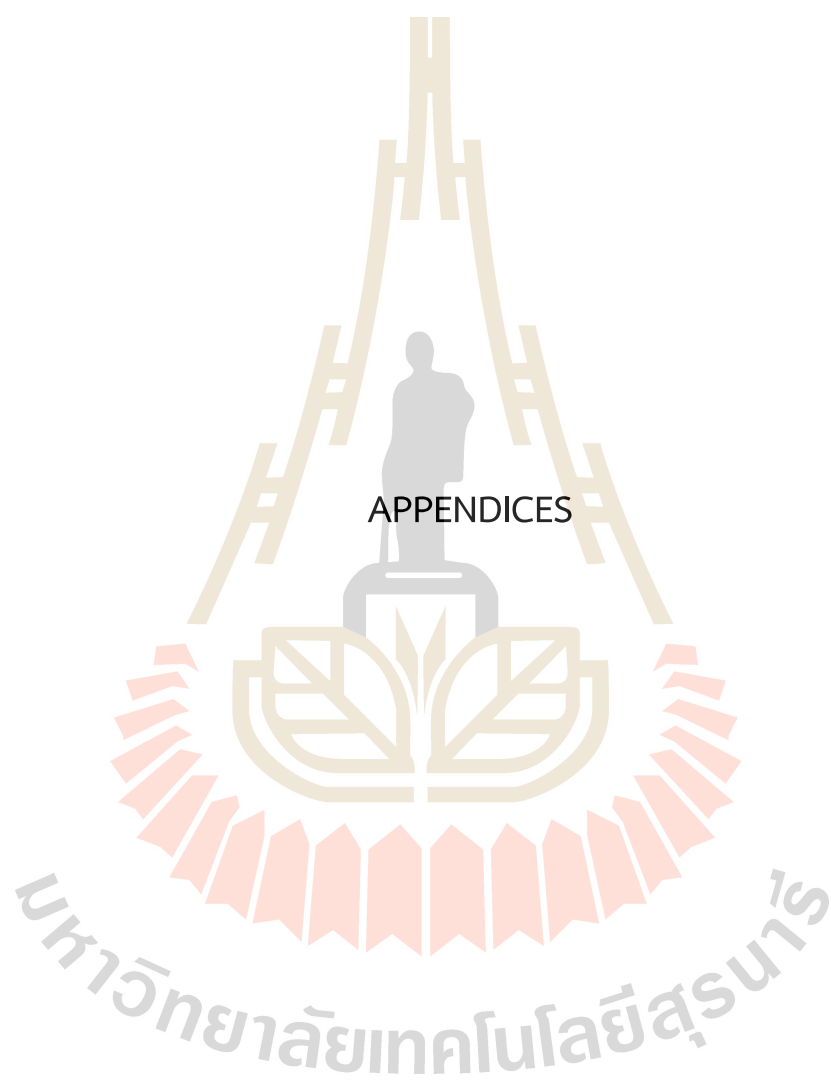
- Chugh, T., and Jain, A. K. (2019b). OCT Fingerprints: Resilience to Presentation Attacks. *ArXiv, abs/1908.00102*.
- Fercher, A., Drexler, W., Hitzenberger, C., and Lasser, T. (2003). Optical Coherence Tomography—Principles and Applications. *Rep. Prog. Phys*, 66. doi:10.1088/0034-4885/66/2/204
- Fercher, A. F., Hitzenberger, C. K., Kamp, G., and El-Zaiat, S. Y. (1995). Measurement of intraocular distances by backscattering spectral interferometry. *Optics Communications*, 117(1), 43-48. doi:https://doi.org/10.1016/0030-4018(95)00119-S
- Gilpin, L. H., Bau, D., Yuan, B. Z., Bajwa, A., Specter, M., and Kagal, L. (2018). *Explaining explanations: An overview of interpretability of machine learning*. Paper presented at the 2018 IEEE 5th International Conference on data science and advanced analytics (DSAA).
- Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*: MIT Press.
- Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., and Pedreschi, D. (2018). A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, 51(5), 1-42.
- He, K., Zhang, X., Ren, S., and Sun, J. (2016). *Deep residual learning for image recognition*. Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.
- Huang, D., Swanson, E. A., Lin, C. P., Schuman, J. S., Stinson, W. G., Chang, W., . . . Puliafito, C. A. (1991). Optical coherence tomography. *Science*, 254(5035), 1178-1181. doi:10.1126/science.1957169
- Kulkarni, A., Chong, D., and Batarseh, F. (2020). Foundations of data imbalance and solutions for a data democracy. In (pp. 83-106).
- LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436-444. doi:10.1038/nature14539
- Lee, J., Min, D., Kim, J., and Kim, W. (1999). A 600-dpi capacitive fingerprint sensor chip and image-synthesis technique. *IEEE J. Solid State Circuits*, 34, 469-475.

- Li, Q., Yang, Y., Guo, Y., Li, W., Liu, Y., Liu, H., and Kang, Y. (2021). Performance Evaluation of Deep Learning Classification Network for Image Features. *IEEE Access*, *PP*, 1-1. doi:10.1109/ACCESS.2020.3048956
- Liu, X., Zaki, F., Wang, Y., Huang, Q., Mei, X., and Wang, J. (2017). Secure fingerprint identification based on structural and microangiographic optical coherence tomography. *Applied Optics*, *56*(8), 2255-2259. doi:10.1364/AO.56.002255
- Loukadakis, M., Cano, J., and O'Boyle, M. (2018). *Accelerating Deep Neural Networks on Low Power Heterogeneous Architectures*.
- Lu, Y., Assaderagh, F., Daneman, M., Jiang, X., Lim, M., Li, X., . . . Boser, B. (2016). *11.2 3D ultrasonic fingerprint sensor-on-a-chip*.
- Lundberg, S. M., Erion, G., Chen, H., DeGrave, A., Prutkin, J. M., Nair, B., . . . Lee, S.-I. (2020). From local explanations to global understanding with explainable AI for trees. *Nature machine intelligence*, *2*(1), 56-67.
- Lundberg, S. M., Erion, G. G., and Lee, S.-I. (2018). Consistent individualized feature attribution for tree ensembles. *arXiv preprint arXiv:1802.03888*.
- Lundberg, S. M., and Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in neural information processing systems*, *30*.
- Memon, S., Sepasian, M., and Balachandran, W. (2009). *Review of finger print sensing technologies*.
- Molnar, C. (2020). *Interpretable machine learning*: Lulu. com.
- Moolla, Y., Singh, A., Saith, E., and Akhoury, S. (2015). *Fingerprint Matching with Optical Coherence Tomography*.
- Perez, L., and Wang, J. (2017). The Effectiveness of Data Augmentation in Image Classification using Deep Learning.
- Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). " *Why should i trust you?*" *Explaining the predictions of any classifier*. Paper presented at the Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining.

- Ridha Ilyas, B., Beladgham, M., Merit, K., and taleb-ahmed, A. (2019). Illumination-robust face recognition based on deep convolutional neural networks architectures. *Vol 18*, 1015~1027. doi:10.11591/ijeecs.v18.i2.pp1015-1027
- Rogowska, J., and Brezinski, M. E. (2000). Evaluation of the adaptive speckle suppression filter for coronary optical coherence tomography imaging. *IEEE Trans Med Imaging*, 19(12), 1261-1266. doi:10.1109/42.897820
- Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2017). *Grad-cam: Visual explanations from deep networks via gradient-based localization*. Paper presented at the Proceedings of the IEEE international conference on computer vision.
- Simonyan, K., Vedaldi, A., and Zisserman, A. (2013). Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.
- Simonyan, K., and Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR*, abs/1409.1556.
- Soifer, V., Kotlyar, V., Khonina, S., and Skidanov, R. (1998). Optical-digital methods of fingerprint identification. *Optics and Lasers in Engineering*, 29(4), 351-359. doi:https://doi.org/10.1016/S0143-8166(97)00122-X
- Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the Inception Architecture for Computer Vision. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2818-2826.
- Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). *Rethinking the inception architecture for computer vision*. Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.
- Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. (2014). How transferable are features in deep neural networks? *Advances in neural information processing systems*, 27.
- Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., and Torralba, A. (2016). *Learning deep features for discriminative localization*. Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.

- Zhu, Y., and Huang, C. (2012). An Improved Median Filtering Algorithm for Image Noise Reduction. *Physics Procedia*, 25, 609-616.
doi:<https://doi.org/10.1016/j.phpro.2012.03.133>
- Zilouchian, A., and Jamshidi, M. (2001). *Intelligent control systems using soft computing methodologies*: CRC press.





APPENDIX A

A PYTHON CODE USED IN THIS THESIS

The code imports the following libraries and modules: TensorFlow (as tf), NumPy (as np), OpenCV (as cv2), os, tqdm, Keras, Matplotlib.pyplot (as plt), scikit-learn, plotly, seaborn, and pandas. These libraries provide various functionalities for deep learning model creation, data manipulation, visualization, performance evaluation, and result analysis.

```
import tensorflow as tf
import numpy as np
import cv2
import os
from tqdm import tqdm #checking progress
from tensorflow import keras
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout, Activation, Flatten
from tensorflow.keras.layers import Conv2D, MaxPooling2D
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from os import listdir
from os.path import isfile, join
from tensorflow.keras.models import load_model
from sklearn.metrics import roc_curve, auc
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
import plotly
import plotly.figure_factory as ff
import seaborn as sn
```

```
import pandas as pd
import numpy as np
from tensorflow.keras.callbacks import CSVLogger
import shap
```

This section determines the size and location of the image.

```
width = 128
num_classes = 12
data_path = 'Identi/train_jpg2/'
data_img = [data_path + f for f in listdir(data_path) if listdir(join(data_path, f))]
data_img
```

The function essentially loads and processes the images from the specified path, assigns one-hot encoded labels based on the folder names, and returns the processed images and labels as lists.

#Convert image to array data and labelling one hot encoding

```
def img2data(path):
    rawimgs = []
    labels = []
    c = 0
    for imagePath in (path):
        for item in tqdm(os.listdir(imagePath)):
            file = os.path.join(imagePath, item)
            c+=1
            l = imagePath.split('/')[2]
            if l == 'Aom1':
                labels.append([1,0,0,0,0,0,0,0,0,0,0,0])
            elif l == 'Aom2':
                labels.append([0,1,0,0,0,0,0,0,0,0,0,0])
            elif l == 'Aom9':
```



```

        labels.append([0,0,1,0,0,0,0,0,0,0])
elif l == 'Aom10':
    labels.append([0,0,0,1,0,0,0,0,0,0])
elif l == 'New1':
    labels.append([0,0,0,0,1,0,0,0,0,0])
elif l == 'New2':
    labels.append([0,0,0,0,0,1,0,0,0,0])
elif l == 'New9':
    labels.append([0,0,0,0,0,0,1,0,0,0])
elif l == 'New10':
    labels.append([0,0,0,0,0,0,0,1,0,0])
elif l == 'Nuvo1':
    labels.append([0,0,0,0,0,0,0,0,1,0])
elif l == 'Nuvo2':
    labels.append([0,0,0,0,0,0,0,0,0,1])
elif l == 'Nuvo9':
    labels.append([0,0,0,0,0,0,0,0,0,0,1,0])
elif l == 'Nuvo10':
    labels.append([0,0,0,0,0,0,0,0,0,0,0,1])
img = cv2.imread(file , cv2.COLOR_BGR2GRAY)
img = cv2.resize(img ,(width,width))
rawimgs.append(img)
return rawimgs, labels

```

These steps prepare the data for training and testing by splitting it, converting it into NumPy arrays, and normalizing the pixel values to a suitable range for the neural network model.

```

x, y = img2data(data_img)
x_train,x_test2,y_train,y_test2 = train_test_split(x,y,test_size = 0.4)
len(x_train), len(x_test)

```

```
x_train = np.array(x_train)
y_train = np.array(y_train)
x_test2 = np.array(x_test2)
y_test2 = np.array(y_test2)
x_train = x_train.astype('float32')
x_test2 = x_test2.astype('float32')
x_train /= 255
x_test2 /= 255
```

This code snippet demonstrates the process of creating a transfer learning model by using a pre-trained model as the base and adding custom layers on top for the specific classification task.

```
## Create the transfer learning model
base_model = tf.keras.applications.{trained_model}(input_shape=(128,128,3),
                                                    include_top=False,
                                                    weights='imagenet')

base_model.trainable = True
num_classes = 12

model = Sequential([
    base_model,
    Dense(16),
    Flatten(),
    Dense(num_classes, activation='softmax')
])

model.summary()
```

This code snippet demonstrates the construction of a custom deep learning model using a combination of convolutional layers, pooling layers, dropout layers, and fully connected layers.

```

## Custom model
model = keras.Sequential([
    keras.layers.Conv2D(128, (3,3), activation='relu', input_shape=(width, width, 3)),
    keras.layers.MaxPooling2D(pool_size=(2, 2)),
    keras.layers.Dropout(0.4),
    keras.layers.Conv2D(128,(3,3) , activation='relu'),
    keras.layers.MaxPooling2D(pool_size=(2,2 )),
    keras.layers.Dropout(0.4),
    keras.layers.Conv2D(128,(3,3) , activation='relu'),
    keras.layers.Dropout(0.4),
    keras.layers.Dense(16),
    keras.layers.Dropout(0.25),
    keras.layers.Flatten(),
    keras.layers.Dense(num_classes, activation = 'softmax') #softmax for one hot . . # sigmoid
for 0/1
])
model.summary()

```

These parameters and configurations help define how the model will be trained and optimized.

```

## Compile model with set parameters
model.compile(optimizer=tf.keras.optimizers.Adam(lr=0.0001), loss='categorical_crossentropy'
, metrics= ['accuracy'])
batch_size = 64
epochs = 500

```

By fitting the model to the training and validating data.

```

## Fitting and save model
history = model.fit(x_train, y_train ,batch_size=batch_size, epochs=epochs
,validation_data=(x_test, y_test), callbacks=[csv_logger])

```

```

model.save(f'Finger_new/model_{nameM}_W{width}_B{batch_size}_Lr{lr}.h5')
print(f'model_{nameM}_W{width}_B{batch_size}_Lr{lr}')

```

Plot the learning curve for basic evaluation.

```

## Plot acc-loss learning curve
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'validation'], loc = 'upper left')
plt.savefig(f'Finger_new/{nameM}_acc_L{lr}_W{width}_B{batch_size}.png', dpi=300)
plt.show()

#Loss
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'validation'], loc = 'upper left')
plt.savefig(f'Finger_new/{nameM}_loss_L{lr}_W{width}_B{batch_size}.png', dpi=300)
plt.show()

## Load 5 trained model
model1 = load_model('Finger_new/model_Our_W128_B64.h5')
model2 = load_model('Finger_new/model_Res50_W128_B64.h5')
model3 = load_model('Finger_new/model_Xcep_W128_B32.h5')
model4 = load_model('Finger_new/model_Vgg16_W128_B64.h5')
model5 = load_model('Finger_new/model_IncV3_W128_B64.h5')

```

Plot the Learning curve of 5 models by loop through the models

```
models = ['Custom', 'ResNet50', 'Xception', 'Vgg16', 'InceptionV3']
```

```
for i, history in enumerate(histories):
```

```
    # create a new figure for each model
    plt.figure()
    # plot the training and validation accuracy
    plt.plot(history['accuracy'], label='Training accuracy')
    plt.plot(history['val_accuracy'], label='Validation accuracy')
    plt.title(f'Model {i} accuracy')
    plt.xlabel('Epochs')
    plt.ylabel('Accuracy')
    plt.legend()
    # plot the training and validation loss
    plt.plot(history['loss'], label='Training loss')
    plt.plot(history['val_loss'], label='Validation loss')
    plt.title(f'{models[i]} model')
    plt.xlabel('Epochs')
    plt.ylabel('Acc-Loss')
    plt.legend()
    plt.savefig(f'Finger_new/History/{models[i]} model2.png', dpi=600)
    plt.show()
```

Plot confusion matrix

```
predict_x={trained_model}.predict(x_test)
```

```
classes_x=np.argmax(predict_x,axis=1)
```

```
classes_x
```

```
y_test_arg=np.argmax(y_test,axis=1)
```

```
print(f'Confusion Matrix_{nameM}_{width}_{batch_size}_V3')
```

```
print(confusion_matrix(y_test_arg, classes_x))
```

```
cm = confusion_matrix(y_test_arg, classes_x)
```

```

cm
labels = ['User1','User2','User3','User4','User5','User6','User7','User8','User9',
'User10','User11','User12']

```

Function for visualize confusion matrix plot with classification report.

```

def cm_plot(cm, labels):
    plt.figure(figsize=(8, 6)) # set the figure size to 8x6 inches
    sn.set_style("whitegrid")
    sn.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=labels, yticklabels=labels)
    plt.xlabel('Predicted Value')
    plt.ylabel('Real Value')
    plt.title(f'Confusion Matrix for {nameM}')
    plt.tight_layout() # adjust the margins to fit the labels within the plot area
    plt.show()

```

```

cm_plot(cm, labels)
report = classification_report(y_test_arg, classes_x, target_names=labels, digits=4)

print(report)

```

Plot ROC and AUC curve for evaluation purpose.

```

## Plot the ROC, AUC curve
predicted_score = model4.predict(x_test)
predicted_score.shape, y_test.shape
sn.set_style("whitegrid")

```

```

fig, ax = plt.subplots()
ax.plot([0, 1], [0, 1], '--', color='gray')

```

```

for i in range(predicted_score.shape[1]):

```

```

y_real = y_test[:, i]
y_score = predicted_score[:, i]

fpr, tpr, threshold = roc_curve(y_real, y_score)
auc_score = auc(fpr, tpr)

name = f"{labels[i]}, AUC={auc_score:.4f}"
sn.lineplot(x=fpr, y=tpr, label=name, data=threshold)

plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title(f'ROC Curve and AUC for {nameM}')
plt.show()

```

The code implements a voting method to combine predictions from multiple models. It loads five pre-trained models and makes predictions on a set of input images. The predictions from each model are averaged to obtain the final prediction. The class with the highest probability is determined as the predicted label. The code then calculates and prints the confusion matrix, which shows the performance of the model's predictions compared to the true labels.

```

def voting_method(input_images):
    # Instantiate the individual models
    model1 = load_model('Finger_new/model_Our_W128_B64.h5')
    model2 = load_model('Finger_new/model_Res50_W128_B64.h5')
    model3 = load_model('Finger_new/model_Xcep_W128_B32.h5')
    model4 = load_model('Finger_new/model_Vgg16_W128_B64.h5')
    model5 = load_model('Finger_new/model_IncV3_W128_B64.h5')

    # Make predictions using all 5 models
    prediction1 = model1.predict(input_images)

```

```
prediction2 = model2.predict(input_images)
prediction3 = model3.predict(input_images)
prediction4 = model4.predict(input_images)
prediction5 = model5.predict(input_images)

# Take the average of all 5 predictions
final_prediction = (prediction1 + prediction2 + prediction3 + prediction4 + prediction5) / 5

# Get the class with the highest probability
predicted_classes = np.argmax(final_prediction, axis=1)

return predicted_classes

# Get the true labels and the predicted labels for the test set
y_test_arg = np.argmax(y_test, axis=1)
predicted_labels = voting_method(x_test)

# classes_x=np.argmax(predicted_labels,axis=1)
# classes_x
# Calculate the confusion matrix
confusion_matrix = confusion_matrix(y_test_arg, predicted_labels)

# Print the confusion matrix
print('Confusion Matrix:')
print(confusion_matrix)
```

The code provided generates SHAP plots, which offer insights into the contribution of image features to the model's decision-making process. These plots enhance the

interpretability of the models and can be included in the appendix to provide visual evidence of the models' transparency and trustworthiness.

```

## Model explainable by SHAP

import os
import matplotlib.pyplot as plt
from tensorflow.keras.preprocessing import image
from tensorflow.keras.applications.xception import preprocess_input

base_folder = 'Finger_new/SHAP/'
class_names = ['User1', 'User2', 'User3', 'User4', 'User5', 'User6', 'User7', 'User8', 'User9', 'User10',
               'User11', 'User12']
models = [model1, model2, model3, model4, model5]
model_names = ['Custom', 'ResNet50', 'Xception', 'Vgg16', 'InceptionV3']

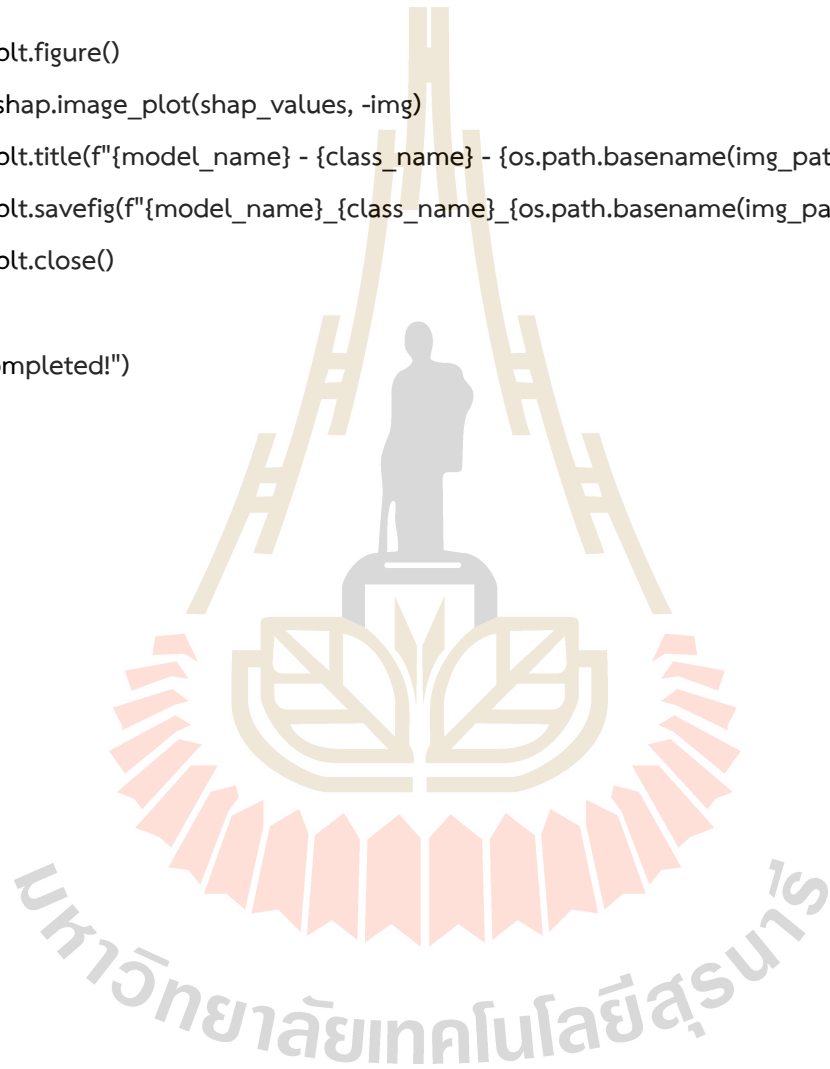
# Function to load and preprocess images
def load_and_preprocess_image(img_path):
    img = image.load_img(img_path, target_size=(128, 128))
    x = image.img_to_array(img)
    x = np.expand_dims(x, axis=0)
    x = preprocess_input(x)
    return x

for class_name in class_names:
    print(f"Processing class: {class_name}")
    class_folder = os.path.join(base_folder, class_name)
    img_paths = [os.path.join(class_folder, img_name) for img_name in os.listdir(class_folder)]

    for img_path in img_paths:
        print(f"Processing image: {img_path}")
        img = load_and_preprocess_image(img_path)

```

```
for model, model_name in zip(models, model_names):  
    background = x_train[np.random.choice(x_train.shape[0], 100, replace=False)]  
    explainer = shap.GradientExplainer(model, background)  
    shap_values = explainer.shap_values(img)  
  
    plt.figure()  
    shap.image_plot(shap_values, -img)  
    plt.title(f"{model_name} - {class_name} - {os.path.basename(img_path)}")  
    plt.savefig(f"{model_name}_{class_name}_{os.path.basename(img_path)}.png")  
    plt.close()  
  
print("Completed!")
```

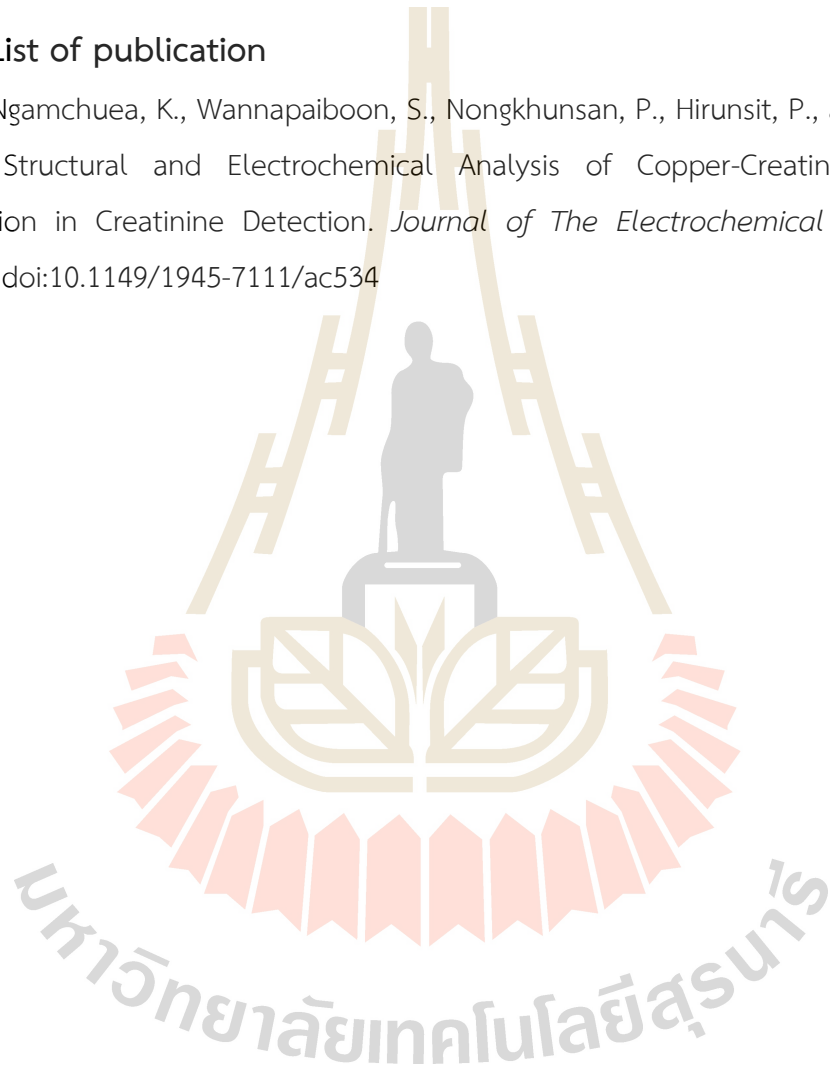


APPENDIX B

PUBLICATION AND PRESENTATION

B.1 List of publication

Ngamchuea, K., Wannapaiboon, S., Nongkhunsan, P., Hirunsit, P., and Fongkaew, I. (2022). Structural and Electrochemical Analysis of Copper-Creatinine Complexes: Application in Creatinine Detection. *Journal of The Electrochemical Society*, 169(2), 020567. doi:10.1149/1945-7111/ac534



Identifying Human Fingerprints by Using Optical Coherence Tomography Image with Deep Learning

Papawit Nongkhunsan^a, Panomsak Meemon^a, Ittipon Fongkaew^{a, 1}

^a*School of Physics, Institute of Science, Suranaree University of Technology, 111 University Avenue, Muang, Nakhon ratchasima, 30000, Thailand.*

Abstract

Optical coherence tomography (OCT) has emerged as a powerful imaging technology for obtaining depth information of living tissue in various fields. In this study, we present a novel approach to enhance the security and performance of traditional fingerprint scanners by leveraging OCT technology for internal finger biometrics. Our proposed system utilizes the intricate internal structure of the fingertip, including sweat glands, epidermis, and dermis, to construct a robust 3D finger model. To extract discriminative features, we employ pre-trained deep learning models such as InceptionV3, VGG16, Xception, ResNet50, and a custom model. Furthermore, to ensure system reliability, we adopt a voting method that combines the predictions of multiple models.

We collect a comprehensive dataset comprising internal fingertips from twelve unique individuals, with approximately 1,000 images per finger. Sixty percent of the dataset is allocated for training, while the remaining portion is used for validation. Through extensive experimentation, we achieve an impressive accuracy and prediction rate of around 99%, as evaluated using the confusion matrix. These findings demonstrate the potential of deep learning-based security systems utilizing internal fingertip data to outperform traditional verification systems and offer enhanced security. Nonetheless, further validation and testing in real-world scenarios are necessary to assess the effectiveness and practical applicability of our proposed method. Future work may involve the integration of this technology with in-house OCT hardware to facilitate seamless implementation in various security applications.

[copyright information to be updated in production process]

Keywords: Optical coherence tomography; Deep learning; Voting method

1. Introduction

In recent years, biometric technology has gained popularity to protect personal information and maintain privacy in numerous businesses. Fingerprint identification is one of the most prevalent biometric techniques. Traditional fingerprint identification methods, however, are susceptible to cyberattacks, particularly those involving forged fingerprints. The vulnerabilities in the technology allow hackers to obtain unauthorized access to sensitive information [1]. Consequently, developing more sophisticated and secure fingerprint identification systems is essential. Conventional fingerprint technology captures external fingertip information using capacitive [2], optical reflection [3], or ultrasonic sensors to generate a template that is compared to the input data when a user tries to access the system [4]. This approach leaves the system susceptible to spoofing attacks by presentation attack instruments (PAIs) that can manipulate the surface of the finger [1]. To protect personal information in the digital age, it is necessary to enhance the security and accuracy of fingerprint identification systems in light of the growing use of biometric technology. Optical coherence tomography (OCT) technology has emerged as a promising solution to the security problems associated with conventional fingerprint identification systems. Similar to ultrasound imaging but with a higher resolution, OCT provides cross-sectional images of living tissues and biological substances using non-harmful near-infrared light [5]. The two most common OCT implementations are TD-OCT and SD-OCT, with the latter being faster and more efficient [6, 7]. Numerous fields, including dermatology [8, 9], ophthalmology [10], biology [11, 12], aquatic toxicology [13, 14], and material characterization [15, 16], have benefited from the application of OCT technology.

Recent research indicates that optical coherence tomography (OCT) technology can significantly improve the security and accuracy of fingerprint identification systems. Nevertheless, traditional methods have flaws, and fake fingerprints can circumvent the technology. The risk stems from the fact that conventional systems

* Corresponding author. Tel.: +66-4422-3705; fax: +66-4422-4185.
E-mail address: ittipon@sut.ac.th

only account for the finger's surface, which is the area that presentation attack instruments (PAIs) can manipulate. PAIs are inexpensive and readily available substances such as glue, play-doh, gelatin, etc. These toys can be used to commit fraud in a traditional high-security system with an approximate 70% success rate [17]. Therefore, developing more sophisticated technologies for detecting and preventing such attacks is essential. The Optical Coherence Tomography (OCT) technology is more effective against attacks involving non-authentic fingerprints, with an actual detection rate (TRD) of 99.73 percent, Chugh et al. [18] reported that OCT could distinguish between spoofed and actual fingers.

In addition, Akbari et al. have investigated the recognition of fingerprints using OCT fingerprint images. Their research demonstrated that the algorithm used to recognize OCT fingerprints could be enhanced by automating fingerprint recognition [19]. Moola et al. have studied traditional fingerprints using OCT and reported that when OCT technology is integrated into the device, the system performs better than normal [20]. To take advantage of the potential of OCT technology, this research aims to develop more secure authentication and verification systems less vulnerable to cyberattacks. We will use SD-OCT to collect and analyze image data in order to identify unique fingerprint images with a 99% degree of accuracy using deep learning techniques and computer vision. We will also perform a classification algorithm on multiple classifications using Python programming. Using OCT images and deep learning techniques, we aim to improve the security and accuracy of fingerprint identification systems in the digital age and safeguard sensitive data.

2. Materials and Methods

2.1 OCT system and data collection

The SD-OCT utilized in this study had a central wavelength of 840 nm and was generated by a superluminescent light-emitting diode, as shown in Figure 1. (SLED). The wave delivers light to the beam splitter, which then separates the light into a reference beam and a sample beam. The reference beam was reflected along the same path as the beam splitter after being directed to a reference mirror. The sample beam is transmitted to the objective lens, which focuses on the sample. Additionally, the objective lens combines the light reflected from the sample's structure. An interference signal was produced when both beams reflected off a beam splitter. A specialized, high-speed spectrometer was utilized to detect the interference signal in the spectral domain. To receive the depth signal, the spectral interference was Fourier transformed. By laterally scanning the focused beam with a Galvanometer mirror, we could acquire a cross-sectional image of 3D data. Each cross-section has around 1000x1000 pixel resolution. The capture area is approximately 4x4 mm, and the sampling distance between each pixel is about 4 mm. The imaging speed is around 25 frames per second.

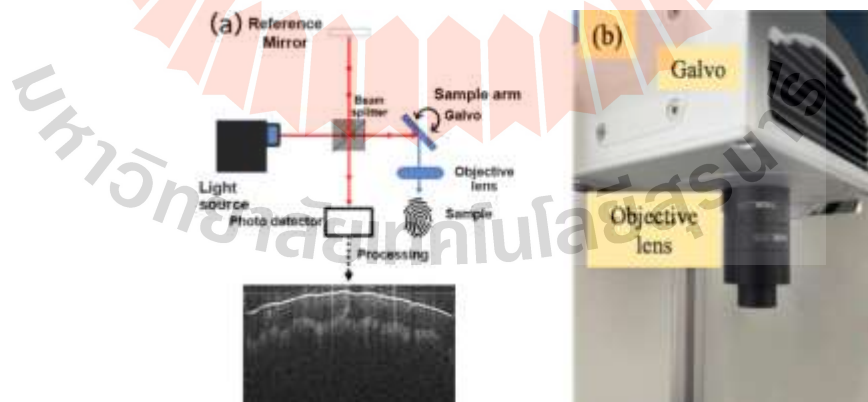


Figure 1. (a) A diagram of the SD-OCT and (b) a sample head for imaging.

2.2 Capturing and collecting internal data from fingertip

We captured internal fingerprint images using in-house OCT hardware with an imaging resolution around 1000x1000 pixels, corresponding to a 4 mm x 4 mm fingerprint area. Figure 2 shows an example of a B-scan

(slice) image. Images reveal that the internal structure of the finger consists primarily of three layers: 1) epidermis, 2) dermis, and 3) subcutaneous. The surface finger or stratum corneum is the uppermost layer of the epidermis, which contains sweat glands [21]. The viable epidermis, which represents the innermost layer of the fingertip, is the subsequent layer. In the dermis layer, the papillary and reticular dermis were visible. As a result, when a user sustains a minor injury, all depth information could serve as a mother template and contain sufficient data for security, even if the finger surface is contaminated because internal information also retains the ridges and valleys. Sweat glands and pores could improve the accuracy of fingerprint identification [22]; this application of OCT properties is a breakthrough and superior to conventional fingerprints, as the latter cannot reach this layer.

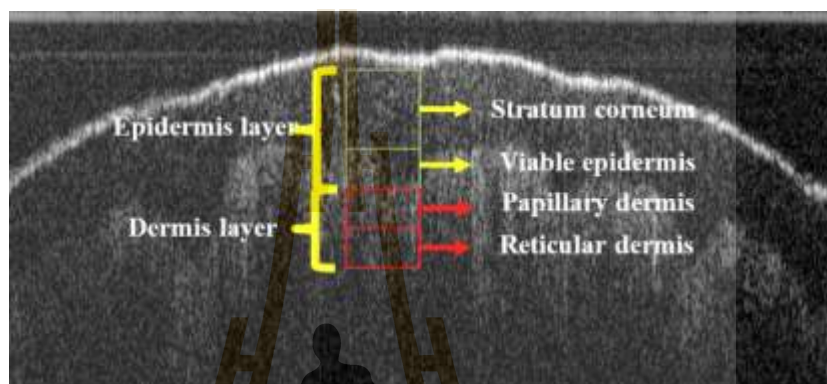


Figure 2. B-scan image with the dept information of fingertip.

According to the aforementioned images, the captured images demonstrated promising performance at multiple depths. Consequently, images are commonly referred to as layer images; this made the oct-captured technique unique because they contain internal information about the finger that should result in a more secure system. As a result, we collected 12 unique fingerprints from three SD-OCT participants; due to the slow scanning speed, each finger must remain still for approximately one minute. From the 3D OCT volume data, we will then extract the 2D fingerprint image. The OCT scanning procedure involves projecting a light beam onto the tissue and measuring the reflected light to generate an image. In the case of internal fingerprint imaging, light is projected onto the fingertip, and the reflected light captures the finger's internal structure, including the sweat glands, epidermis, and dermis. The resulting OCT image can be utilized to extract fingerprint identification features, providing a potentially more secure alternative to conventional fingerprint scanners.

2.3 Image augmentation

Image augmentation is used to randomly adjust the dataset, thereby increasing the dataset's diversity for a more accurate model and mitigating the overfitting issue. The outcome of the prediction should be better. Image processing enlarges a small dataset; examples of image processing include adjusting contrast, and noise adjustment. The effect of image enhancement is enhancing the performance of the model [23]. An example of the image being enlarged is shown in Figure 3.

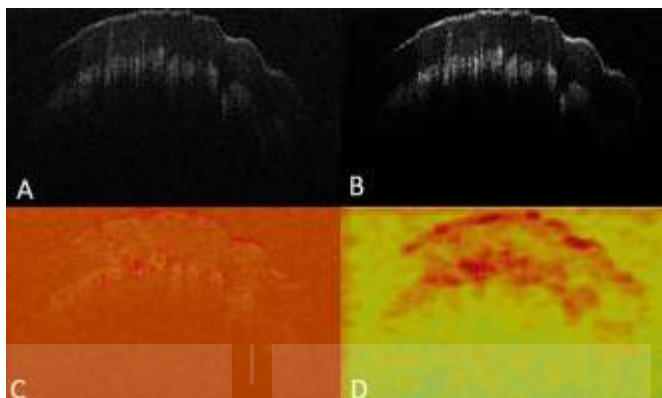


Figure 3 Image processing which A)raw data, B) contrast adjust with noise filter (median filter), C) and D) Heat mapping

Figure 3 depicts the image processing of an OCT fingerprint, where A is the raw data, and B is the processed image with noise (median filter) and contrast adjustments. C and D are heat maps used to comprehend the information on the various sides and how networks will extract features from the images. As can be seen, image processing would provide significantly more information about internal fingertip data. Figures 3C and 3D show that the interior surface has a high density; we could use this region as a biomarker for fingerprints [24]. With all the interior raw and processed data details, it is possible to predict what computers and convolutional neural networks (CNNs) will look for in this data. After analyzing the OCT-obtained internal fingerprint images, we observed increased information density from the surface to the interior.

Deep learning techniques can improve the accuracy of fingerprint recognition by incorporating this information. Figure 3A depicts the unprocessed OCT image of the fingerprint. It is difficult to extract useful characteristics from it. As demonstrated in Figure 3B, we applied a median filter to the image to reduce noise and extract more distinct features. Using the OCT data, we also generated a heat map of the fingerprint, as shown in Figures 3C and 3D. The heat map depicts areas of high and low image intensity. We can observe that high-density regions are present from the fingerprint's surface to its interior. This data confirms that it can be used to create a more precise and secure fingerprint identification system. We use heat mapping to ensure that OCT fingerprint images contain sufficient internal information for identification. Only The images were processed with a median filter (3B), and our CNN was fed the raw data (3A). High levels of noise and interference in the raw OCT images can negatively impact the performance of the CNN. To solve this issue, we used a median filter to reduce noise and preserve the fine details of the fingerprint valleys, ridges, and other information.

2.4 Deep Learning and Classification Model

Our deep learning strategy utilized convolutional neural networks in this work. The dataset was split into two sets: a training set and a validation set, with 60% of the data assigned to the training set and 40% to the validation set. For the multiclass labelling of 12 fingers, one-hot encoding was used to refer to 12 unique users, the reason for 12 classes due to the limitation of graphics processing unit's memory is 6 GB. We downsized the images and converted them to arrays to conserve computational resources. The initial layer of our model was a convolutional layer that extracted image features. Next, the data was sent to the max pooling layer, which reduced the data's size while retaining only the most essential details or values. We utilized the dropout layer to prevent overfitting [25]. The activation function rectified linear unit (ReLU), was essential to the design of our neural network architecture.

2.4.1 Architecture of model

In this work, the model was created from five different architectures. In the first design, we will design our custom network architecture for predicting the fingerprint. In the other models, we will use the complex architecture from ImageNet that InceptionV3, Resnet50, VGG16, and Xception will use for training the model, respectively [26, 27]. Then, we will ensure user confidence in the system by combining all three models. For the design parameter depicted in Figure 4, a convolutional layer with 128 features and a kernel

size of 3×3 was extracted. Therefore, the ReLu activation function was utilized in this layer [28]. The pooling layer utilized max-pooling to collect the 2×2 essential data. A learning rate of 0.0001, a batch size of 128, and 1000 epochs were used during training. In this work, the models were compiled using the Adam optimizer, a batch size of 64, and 500 training iterations. After each training cycle, the learning rate was decreased by a factor of 10 from 0.1 to 0.000001 in order to promote convergence. However, due to out-of-memory (OOM) issues, it could not further reduce the learning rate. Due to OOM constraints, the maximum batch size was limited to 64 units. Due to OOM issues, the input image size was also restricted to $128 \times 128 \times 3$. It was impossible to reduce the model's size any further because doing so would have necessitated the removal of essential components and negatively impacted its performance.

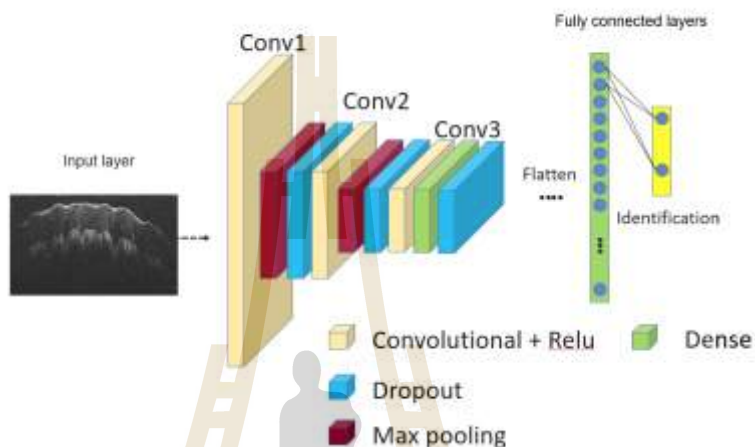


Figure 4 Neural network architecture diagrams of a model

Google developed the InceptionV3 architecture as the third iteration of its deep learning convolutional neural network architecture for image classification and identification applications. InceptionV3 was used to train around one thousand classes using an ImageNet picture dataset containing over one million images. This InceptionV3 has a high classification efficiency; hence it was chosen for fingerprint training [27]. The following model selected is the Xception model. The Xception architecture is based on the Inception architecture, which extracts information from input images using a combination of 1×1 and 3×3 convolutional filters. Xception, however, replaces the 3×3 filters with depthwise separable convolutions, which involve two steps: first, applying a single convolutional filter to each input channel (i.e., the "depthwise" part), and then concatenating the results and applying a 1×1 convolutional filter (i.e., the "separable" part) [26]. Next, the ResNet50 model, a deep residual network, was developed and trained on the ImageNet dataset, which consists of millions of photographs of various objects. The massive dataset from which our pre-trained model has acquired a wealth of features can serve as the basis for our fingerprint classification problem. The pre-trained ResNet50 model will be fine-tuned by training it on a smaller dataset of OCT images of human fingerprints. This fine-tuning procedure will update the pre-trained model's weights [29]. Significant image classification dataset ImageNet was previously used to train the deep convolutional neural network model with the VGG16 architecture. Transfer learning modifies a once-trained model for a new task by adding data, in this case, OCT pictures, into the weights of its layers. Several fully connected layers follow several stacks of convolution and pooling layers in VGG16's architecture. The convolution layers collect spatial and channel-wise data, whereas the pooling layers lower the spatial dimension of the feature maps while preserving the essential data. The layers with complete connectivity create predictions depending on the input image. The output of the VGG16 model may be connected to a fully connected layer containing several neurons equal to the number of classes in the target task, in this case, the number of classes for various fingerprints. The class of an input image is predicted based on the output of the fully linked layer. Therefore, rather than starting from scratch, the modified VGG16 model can be trained end-to-end using the OCT data, thereby increasing its accuracy for the fingerprint detection task [30].

2.5 Ensemble model

Table 1 The parameter settings for four TL networks and a custom network.

We proposed the ensemble model for more precise or reliable forecasting. Using a weighted average or

| Parameters | Initial input size | Initial learning rate | Batch-size | Epoch | Optimizer |
|---------------------|--------------------|-----------------------|------------|-------|-----------|
| Networks | | | | | |
| Inception-V3 | 128x128x3 | 0.00001 | 64 | 500 | Adam |
| Xception | 128x128x3 | 0.00001 | 64 | 500 | Adam |
| VGG-16 | 128x128x3 | 0.00001 | 64 | 500 | Adam |
| Resnet-50 | 128x128x3 | 0.00001 | 64 | 500 | Adam |
| Custom model | 128x128x3 | 0.00001 | 64 | 500 | Adam |

some other mathematical function, an ensemble model combines the aforementioned distinct models to provide a single output. Typically, the ensemble model is utilized in applications where the objective requires the highest achievable level with limited computational resources and data. The ensemble model we employed is a voting method classifier, a voting ensemble in which each model makes a prediction, and a majority vote determines the final prediction. For instance, if five models forecast A, B, and A, the final prediction would be A. In our scenario, there are twelve classes for each finger. Figure 5 depicts the diagram of the voting ensemble model. We proposed the ensemble model for more precise or reliable forecasting. Using a weighted average or some other mathematical function, an ensemble model combines the aforementioned distinct models to provide a single output. Typically, the ensemble model is utilized in applications where the objective requires the highest achievable level with limited computational resources and data. The ensemble model we employed is a voting method classifier, a voting ensemble, in which each model makes a prediction, and a majority vote determines the final prediction. For instance, if five models forecast A, B, and A, the final prediction would be A. In our scenario, there are twelve classes for each finger. Figure 5 depicts the diagram of the voting ensemble model.

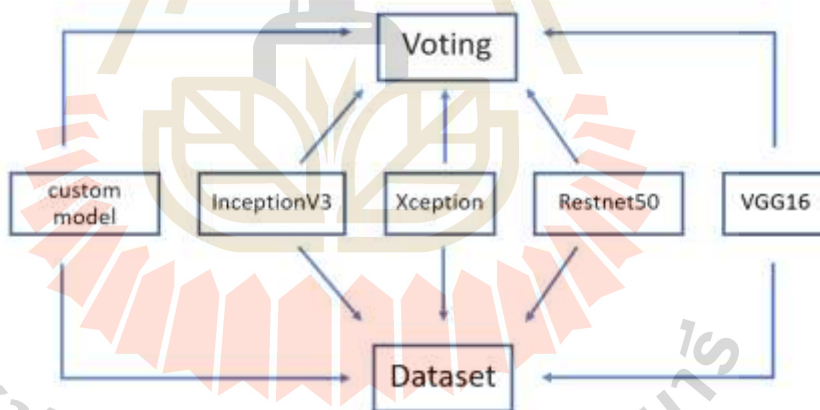


Figure 5 diagram of voting ensemble method

The parameters utilized by the networks are detailed in Table 1. Set the initial input size and batch size to optimize the use of computational resources. The learning rate is then modified until the learning curves converge, ensuring the model can generalize well to the current dataset.

The model weights were modified during training using backpropagation to minimize the categorical cross-entropy loss function. The models were created utilizing the Adam optimizer, 64-person batches, and 500 training iterations. After each training cycle, the learning rate was reduced by 10 from 0.1% to 0.000001% to promote convergence. It was unable to cut the learning rate further due to out-of-memory (OOM) difficulties. Due to OOM restrictions, the batch size could not exceed 64, and the maximum image size was 128x128x3. It was impossible to reduce the size any further because doing so would have resulted in losing important components and diminished the model's performance.

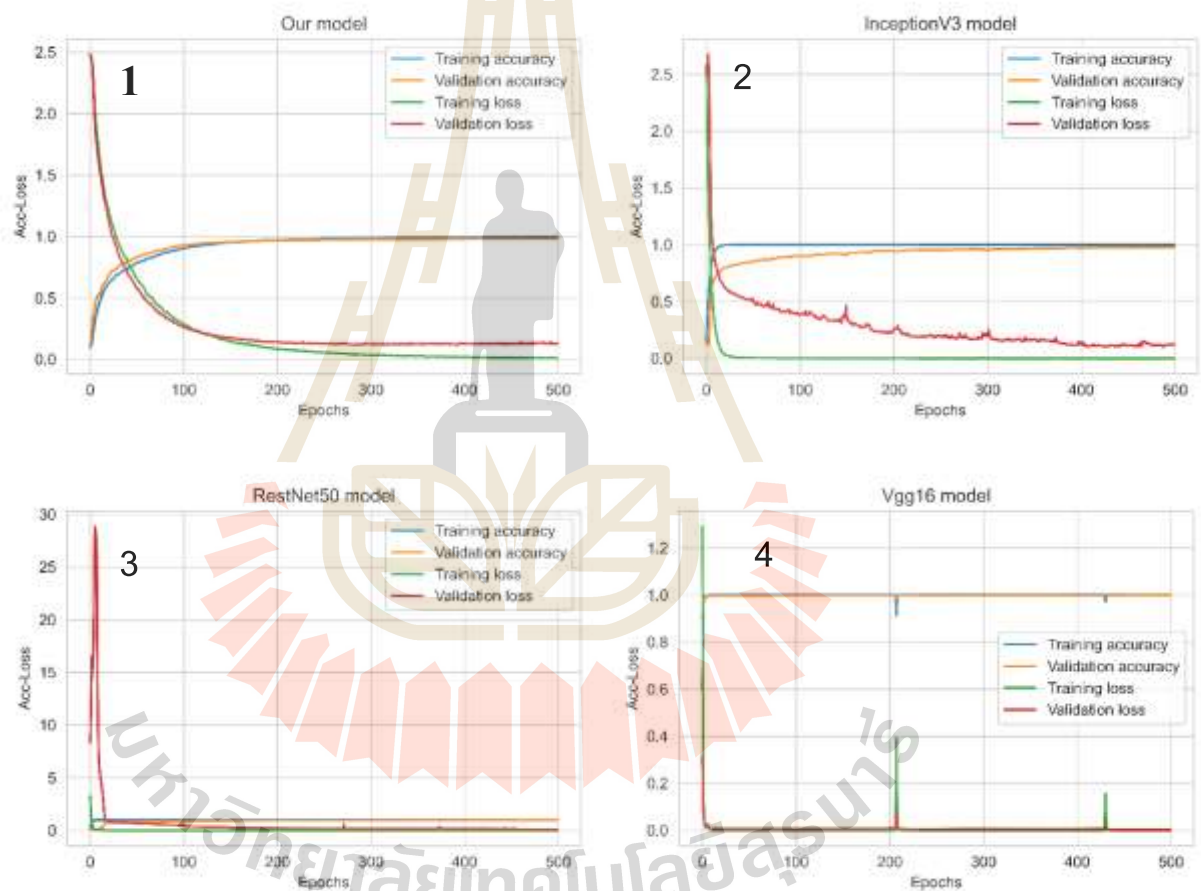
2.6 Evaluation model

Using model evaluation, we can determine how well the model can solve our problem. In addition, various metrics can be used to evaluate the performance of a DL model. These metrics depend on the task the model was designed for and can include accuracy, precision, recall, and F1 score. Then we will use it to calculate and visualize the model's performance through a learning curve and confusion matrix [31].

3. Results

3.1 Learning curve

This work uses images of twelve internal fingertips of three people that capture from OCT scans to identify the person using deep learning techniques. We create a dataset for around 12,108 impressions. Then we randomly split 60 percent for the training set and 40 percent for the validating set for each model. The learning curve of each model will show in Figure 6 below.



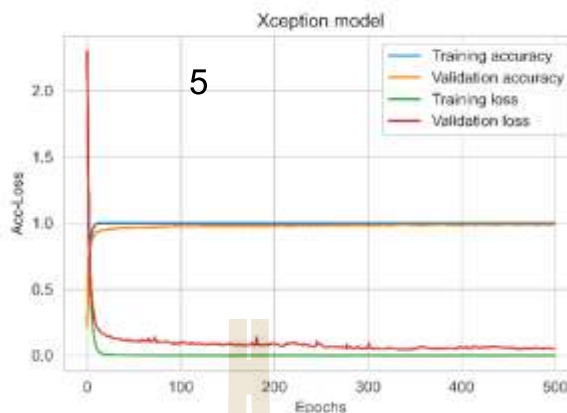
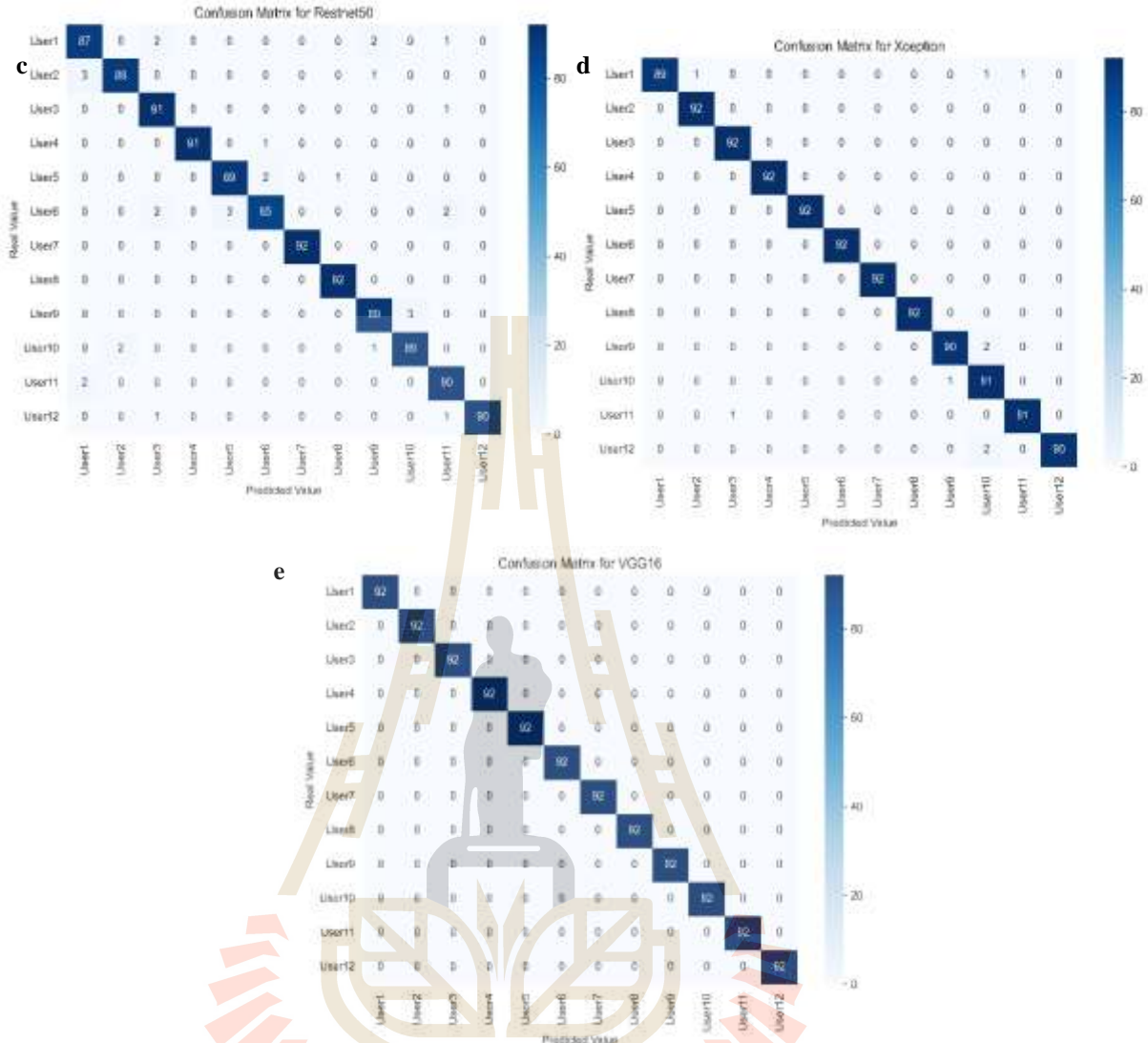


Figure 6 The learning curves or accuracy and loss curves of five models 1) Individual model, 2) InceptionV3 model, 3) ResNet50 model, 4) Vgg16, and 5) Xception model.

The learning curve is important when assessing a machine learning model's performance. It gives a broad picture of the link between the model's accuracy and loss and the total number of training epochs. The training accuracy and loss are metrics used to evaluate how well the model performs during training on the training dataset. The training accuracy is the percentage of correctly classified images in the training dataset, while the training loss measures how well the model is able to predict the correct label for each image in the training dataset. The model's accuracy generally grows while the loss decreases as the number of epochs rises. This pattern might not always hold, and the learning curve can be used to spot problems with the model, including overfitting, underfitting or lack of convergence. To optimize the performance of the model and adjust its hyperparameters, it is crucial to analyse the learning curve.

Figure 6 displays the result of training the data. First is the custom model result (Figure 6.1); training and validation accuracy started at low values and increased as the number of epochs increased. The training and validation loss, on the other hand, started high and decreased as the number of epochs increased. The initial high loss could be due to the random initialization of the model's weights. The weights were optimized as the model continued to train, resulting in a decreased loss. The validation accuracy also increased along with the training accuracy, indicating that the model was not overfitting to the training data. Additionally, the validation loss started higher than the training loss and remained slightly higher throughout training, which could be because the validation data was not used for weight optimization.

Based on the learning curve for the InceptionV3 model in (Figure 6.2), we observed that the accuracy and validation accuracy started at a higher initial value compared to the first model (Figure 6.1). The validation accuracy for this model was relatively consistent and had a slight dip towards the end. The training and validation loss showed a similar pattern as our model, where it started high and converged towards zero with higher epoch values. However, the final training loss was lower in this model, which indicates that the model could fit the data more accurately. Overall, the learning curve for the second model showed that the model was able to perform well on the data and achieve high accuracy with low loss values. Figure 6.3 illustrates the learning curve for the third model, ResNet50, which began with low training and validation accuracy and significant training and validation loss. The model quickly incorporated additional data, and by epoch 8, it had an accuracy of 0.99 and a loss of 0.034. The subsequent rapid increase in training and validation losses implies that the model may have begun to overfit. Despite this, the model completed the training phase with a high degree of accuracy. With a final score of 0.97 and a validation loss of 0.10, validation accuracy maintained a high level of consistency. Despite some evidence of overfitting, the ResNet50 model performed well in classifying fingerprint images accurately. Figure 6.4 depicts the learning curve for the VGG16 model. From the beginning epochs, both the training and validation accuracy grow dramatically while the loss decreases. At epoch 207, the model reaches its highest level of accuracy, with a training accuracy of 0.91, a loss of 0.39, a validation accuracy of 0.98, and a validation loss of 0.07. After the peak, the accuracy does not increase appreciably, however, the validation loss grows substantially at later epochs, indicating overfitting. The model achieved great accuracy on both the training and validation sets, with a final validation accuracy of 0.99 and a final training and validation loss of 0.002. In Figure 6.5, the learning curve for the Xception model shows a slow start with low accuracy and high loss in the beginning. However, as the number of epochs increases, the model converges, and the training and validation accuracy improves significantly



From Table 2a, Custom model: The large number of accurate predictions (90 or more) for the majority of classes demonstrates the custom model's high overall accuracy. Some things could be improved in the predictions, particularly for users 1, 6, and 11, where the model misclassified one or two data. For user 4, the model has the highest accuracy, with every prediction coming true. Most of the InceptionV3 (Table 2b) model's predictions were accurate, showing a high level of overall accuracy. Each of users 1, 3, 5, and 11 has a number of erroneous predictions, indicating that the model has difficulty with these people. Users 4 and 7 have the most accurate model, with all accurate forecasts. For Table 2c ResNet50 model, the majority of predictions were accurate, suggesting a good level of general accuracy. The model has difficulty with users 1, 3, 5, and 10, as each of these classes has some incorrect predictions. Users 4, 7, and 8 have the most accurate model, with all accurate forecasts. In addition, Table 2d demonstrates that the VGG16 model has a high overall accuracy, with the majority of forecasts being true. The algorithm has difficulty with users 1, 5, and 11, as each of these categories has several erroneous predictions. For user 4, the model's accuracy is the highest, since all predictions are accurate. The Xception model has the highest overall accuracy for Table 2e and produces accurate predictions for the majority of courses. There are a few mistakes in the forecasts for users 1, 2, 3, and 11. The algorithm misclassified one or two data points as belonging to a different user. For user 4, the model's accuracy is the highest, since all predictions are accurate.

The confusion matrix, in short, offers a helpful summary of each model's performance, showing each strategy's advantages and disadvantages—the result after testing five models with unseen 1104 images. It specifically focuses on True Positives (TP) and False Negatives (FN), showing the accuracy of predictions

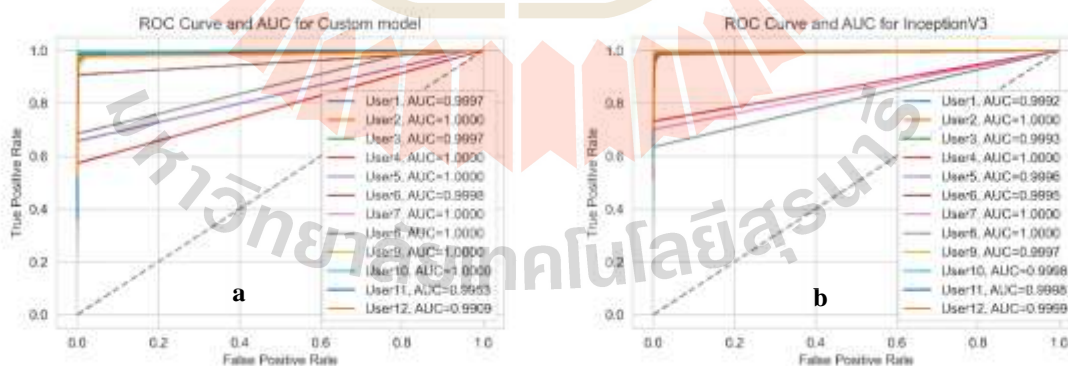
and instances where the model misclassified data. The overall classification performance can be enhanced using these results to inform future model or data pre-processing adjustments. In summary, precision, recall, and F1 score are evaluation metrics calculated from a confusion matrix and used to assess a classifier's performance, illustrated in Table 3.

Table 3 Classifier report table.

| Network | Accuracy | Precision | Recall | F1-score | Support |
|---------------|----------|-----------|--------|----------|---------|
| Inception-V3 | 97.59 | 97.65 | 97.61 | 97.62 | 1104 |
| Xception | 99.13 | 99.2 | 99.18 | 99.19 | 1104 |
| VGG-16 | 100 | 100 | 100 | 100 | 1104 |
| Restnet-50 | 97.1 | 97.18 | 97.1 | 97.12 | 1104 |
| Custom model | 98.65 | 98.66 | 98.69 | 98.67 | 1104 |
| Voting method | 100 | 100 | 100 | 100 | 1104 |

Table 3 displays the performance result of four different pre-trained models, a custom model, and a voting method. The metrics include accuracy, precision, recall, and F1-score. The models achieved high levels of accuracy, ranging from 97.1% to 100%. The VGG-16 model achieved 100% accuracy, correctly classifying all the instances in the test set. The Xception model achieved the highest accuracy of 99.13%, followed by the custom model with an accuracy of 98.65%. In terms of precision, recall, and F1-score, all models performed well, with scores ranging from 97.18% to 100%. The VGG-16 model achieved perfect scores of 100% for all metrics, while the custom model achieved the highest F1 score of 98.67%.

The classification report shows that all models, including the custom model, performed well in classifying the test set. However, it's crucial to remember that measurements like accuracy alone can be deceptive, especially when working with unbalanced datasets. That's why it's essential to consider other metrics, such as AUC and ROC curves, to better understand the overall model performance. In this regard, the pre-trained deep learning models and the custom model achieved high AUC scores, with the voting method combining their strengths to achieve perfect scores in all metrics. Therefore, using multiple models and evaluation metrics, including AUC and ROC, can lead to a more robust and reliable classification performance below in Figure 7(a-f).



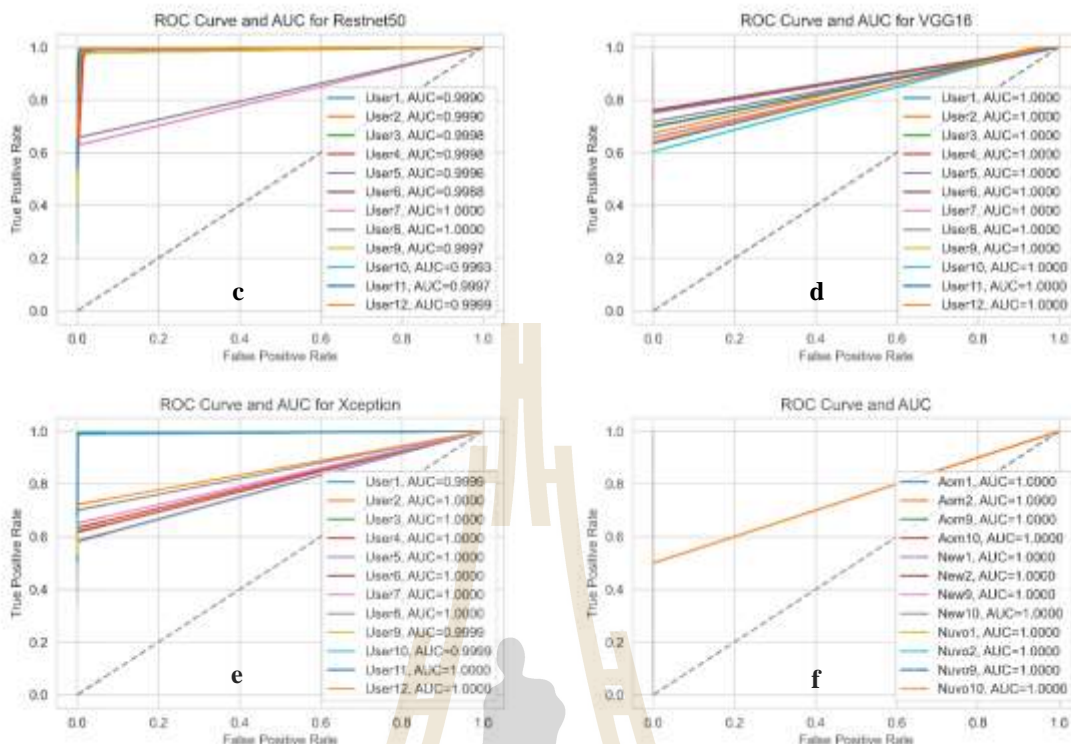


Figure 7 AUC and ROC curve that shows the performance of the 5 models a) Individual model, b) InceptionV3 model, c) RestNet50 model, d) Vgg16, e) Xception model, and f) voting method.

Specifically, our designed model (Figure 7a) for this purpose: The AUC for this model is 0.9953, showing that it distinguishes between positive and negative classifications adequately. The ROC curve illustrates that when the false positive rate (FPR) increases, the true positive rate (TPR) increases rapidly, showing that the model has high sensitivity and can identify real positives at a low cost of false positives. When the TPR reaches 1, the curve flattens, indicating that the model's classification performance at that level is optimal. In Figure 7b, the AUC for the InceptionV3 model is 0.9998, which is greater than the AUC for the custom model and demonstrates that it performs extraordinarily well at differentiating positive and negative classes. The ROC curve illustrates that while the FPR grows, the TPR climbs rapidly, confirming the model's high sensitivity and ability to recognize true positives with a low rate of false positives. When the TPR reaches 1, the curve flattens, indicating that the model's classification performance at that level is optimal, indicating that the model can perform perfectly for classification at that level. Figure 7c demonstrates that the AUC for ResNet50, which is also quite high at 0.9997, distinguishes between positive and negative classifications. The ROC curve illustrates that while the FPR increases, the TPR increases rapidly, suggesting the model's high sensitivity and capacity to recognize true positives with a comparatively low rate of false positives. When the TPR reaches 1, the curve flattens. The models for VGG16 and Xception (Figures 7d and 7e, respectively) exhibit an AUC of 1, the maximum attainable value, demonstrating remarkable ability in distinguishing between positive and negative classifications. The ROC curve illustrates the model's capacity to achieve optimum classification performance, with a TPR of 1 and an FPR of nearly 0, showing that the model has a high degree of sensitivity and specificity. In the final case voting technique (Figure 7f), the AUC is 1.0000, which is the maximum attainable value and indicates that the model distinguishes between positive and negative classifications perfectly. The high TPR values for various thresholds indicate that the model has a high sensitivity, i.e., it properly identifies a substantial number of positive cases. The low FPR values also indicate that the model has a high specificity, which indicates that it properly identifies a substantial fraction of negative situations. The outcomes indicated that the voting mechanism is highly effective for classifying data.

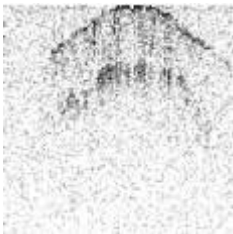
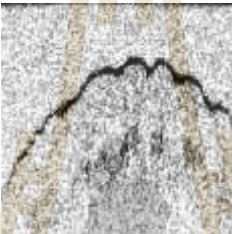
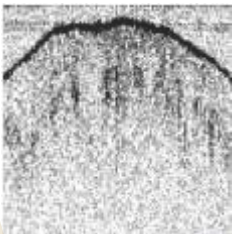
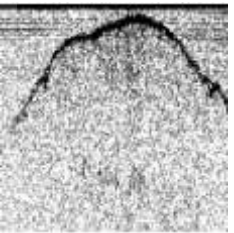





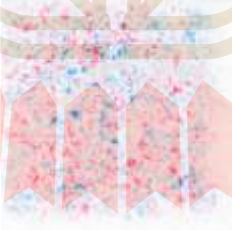






It is clear from examining the outcomes of the many models and the voting process that each model exhibits distinct performance levels, as evidenced by their AUC values and the forms of their individual ROC curves. The distinct qualities and advantages of each model are highlighted by the disparities in the AUC

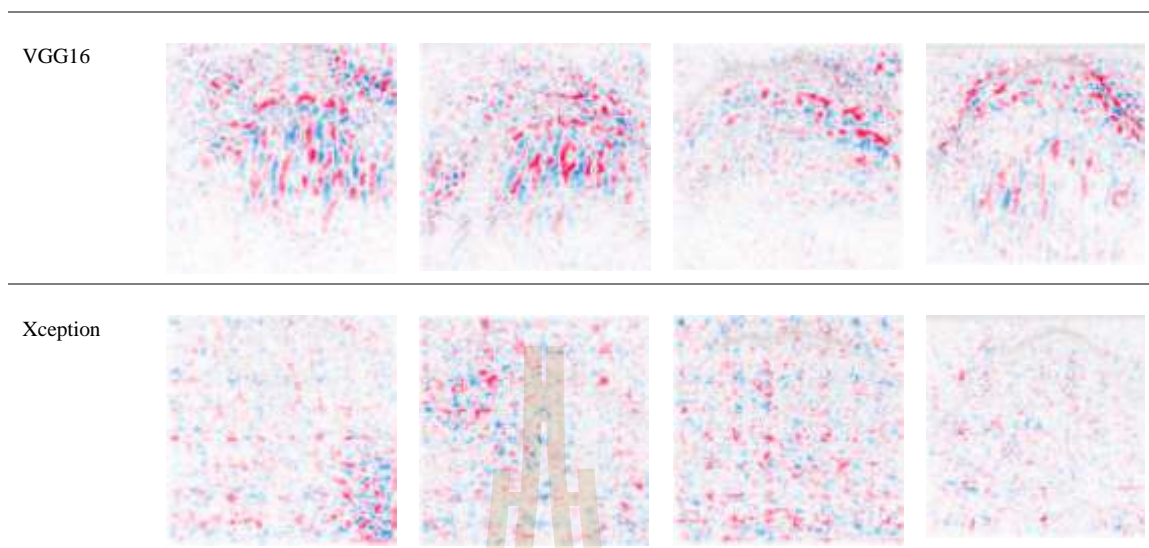
values and ROC curve shapes, which may be traced to the particular behaviour of each classifier for the given task. It is important to consider not only the AUC values and ROC curves but also other performance metrics such as precision, recall, and F1-score, to gain a comprehensive understanding of each classifier's performance. Additionally, analysing the confusion matrix can provide valuable insights into the model's overall performance and potential areas for improvement.

3.3 Model interpretation

Due to the growing use of sophisticated AI models and algorithms in crucial decision-making procedures, Explainable Artificial Intelligence (XAI) has attracted much interest in recent years [32, 33]. We use shap known as SHapley Additive exPlanations (SHAP) [34]. SHAP aims to provide understandable explanations for individual predictions made by any machine learning model. The table below will show the explainable model by random image for each model. The result of the model interpretation will show in Table 4.

Table 4 Explainable model table shows the feature selection

| | | | | |
|----------------|---|---|--|---|
| Original image |  |  |  |  |
| Custom model |  |  |  |  |
| InceptionV3 |  |  |  |  |
| ResNet50 |  |  |  |  |



In this study, we analysed the trustworthiness and interpretability of five distinct models: 1) a custom model, 2) InceptionV3, 3) VGG16, 4) Xception, and 5) ResNet50. We used the SHAP method to determine the significance of features and model explanations. It provided valuable insight into the decisions made by these models when presented with random, unseen data. SHAP generates colour-coded descriptions, where each colour represents the impact of a particular feature on the model's prediction. Positive values (red) indicate that the presence of the proportional feature increases the prediction probability, whereas negative values (blue) indicate that the presence of the proportional feature decreases the prediction probability. The impact on the prediction is proportional to the magnitude of the colour's intensity. The following table provides an exhaustive explanation of the procedure for selecting features. We consider four randomly selected images from the unseen dataset to illustrate how each model interprets the input data differently.

Using the SHAP method, the following analysis examines the feature selection procedure for five distinct models: models - Custom, InceptionV3, ResNet50, VGG16, and Xception. We evaluate four randomly selected images from an unseen dataset to illustrate how each model interprets the input data differently. The Custom model focuses primarily on the surface area of the fingertip, whereas the InceptionV3 and ResNet50 models exhibit a wider distribution of influential features throughout the images. In contrast, the VGG16 model focuses on particular, important image features, such as the fingerprint surface area, sweat gland areas, and sweat pores. Lastly, the Xception model exhibits a red cluster distribution similar to that of the InceptionV3 and ResNet50 models. By gaining an understanding of these feature selection processes, we can gain valuable insights into the decision-making mechanisms of each model, which may inform the development of more effective models in future research. By analyzing and comparing the interpretability results of these five models in conjunction with traditional evaluation metrics such as accuracy, precision, recall, and F1-score, as well as domain-specific metrics pertinent to the problem at hand, we can gain a deeper understanding of their behaviour. This holistic approach will ultimately aid in guiding the selection of the most suitable model for a given application and enhancing model performance through the engineering of targeted features.

4. Discussion and conclusion

In this study, a voting ensemble deep learning model was developed for fingerprint classification using optical coherence tomography (OCT) images. The model comprised five models, including an individual model designed by the authors, an InceptionV3 model, a Resnet50 model, a VGG16 model and an Xception model. The model achieved strong performance, as evidenced by the high AUC values on the ROC curves for each class and model. The model achieved an overall AUC of 1 on the validation set and a holdout test set, demonstrating its ability to generalize to unseen data. Using OCT images as input for the ensemble model allowed for capturing high-resolution images of fingerprints, contributing to the model's good performance. The model's performance was further visualized through ROC curves, which showed that the model could effectively distinguish between the different classes of internal fingerprints.

The results of this study indicate that the voting ensemble model based on a convolutional neural network utilizing OCT images is a promising approach for fingerprint classification tasks. However, there is still room for performance enhancement with this model. The model's generalization ability and overall performance may be enhanced by additional hyperparameter tuning and the use of a larger and more diverse training dataset. Using the AUC values on the ROC curves can be a useful metric for assessing the performance of the model in such tasks. Observed differences in AUC values and ROC curve shapes between models highlight the importance of selecting the optimal model for a given task while also considering other performance metrics and classifier properties. As demonstrated by the exceptional AUC value of 1.00 achieved in this instance, it is possible to enhance classification performance by combining the strengths of multiple models using the voting method. This study contributes to the development of effective and trustworthy fingerprint classification methods, which may have applications in forensics and biometric identification.

Acknowledgements

I am deeply grateful to my project advisor, Dr. Ittipon Fongkaew, for his continuous support and guidance in the programming aspects of this research. His expertise and commitment have been instrumental in the successful completion of this work. My sincere appreciation goes to my co-advisor, Assoc. Dr. Panomsak Meemon, and his team at the School of Physics, Institute of Science, Suranaree University of Technology, for their invaluable contributions in handling the OCT hardware and image data. Their knowledge and assistance greatly enriched this study.

I would also like to extend my gratitude to my colleagues and fellow researchers at Suranaree University of Technology for their collaboration and stimulating discussions, fostering my growth as a scholar and enriching my research experience.

References

1. Biggio, B., et al., Security evaluation of biometric authentication systems under real spoofing attacks. *IET Biometrics*, 2012. 1(1).
2. Lee, J., et al., A 600-dpi capacitive fingerprint sensor chip and image-synthesis technique. *IEEE J. Solid State Circuits*, 1999. 34: p. 469-475.
3. Soifer, V., et al., Optical-digital methods of fingerprint identification. *Optics and Lasers in Engineering*, 1998. 29(4): p. 351-359.
4. Lu, Y., et al., 11.2 3D ultrasonic fingerprint sensor-on-a-chip. 2016.
5. Huang, D., et al., Optical coherence tomography. *Science*, 1991. 254(5035): p. 1178-81.
6. Fercher, A., et al., Optical Coherence Tomography—Principles and Applications. *Rep. Prog. Phys.*, 2003. 66.
7. Cense, B., et al., Ultrahigh-resolution high-speed retinal imaging using spectral-domain optical coherence tomography. *Optics Express*, 2004. 12(11): p. 2435-2447.
8. Lee, K.-S., et al., Cellular resolution optical coherence microscopy with high acquisition speed for in-vivo human skin volumetric imaging. *Optics Letters*, 2011. 36(12): p. 2221-2223.
9. Rolland, J.P., et al., Gabor Domain Optical Coherence Microscopy of Human Skin, in *Advances in Dermatological Sciences*. 2014, The Royal Society of Chemistry. p. 37-52.
10. Wojtkowski, M., et al., Three-dimensional retinal imaging with high-speed ultrahigh-resolution optical coherence tomography. *Ophthalmology*, 2005. 112(10): p. 1734-46.
11. Davis, A.M., et al., In vivo spectral domain optical coherence tomography volumetric imaging and spectral Doppler velocimetry of early stage embryonic chicken heart development. *J Opt Soc Am A Opt Image Sci Vis*, 2008. 25(12): p. 3134-43.
12. Tearney, G.J., et al., Rapid acquisition of in vivo biological images by use of optical coherence tomography. *Optics Letters*, 1996. 21(17): p. 1408-1410.
13. Bellini, N., et al., The Application of Optical Coherence Tomography to Image Subsurface Tissue Structure of Antarctic Krill *Euphausia superba*. *PloS one*, 2014. 9: p. e110367.
14. Thanomsit, C., J. Saetiew, and P. Meemon, Optical Coherence Tomography as an Alternative Tool for Evaluating the Effects of Glyphosate on Hybrid Catfish (*Clarias gariepinus* × *Clarias macrocephalus*). *Toxicology Reports*, 2022. 9.

15. Meemon, P., et al., Optical Coherence Tomography Enabling Non Destructive Metrology of Layered Polymeric GRIN Material. *Scientific Reports*, 2013. 3(1): p. 1709.
16. Palawong, K., et al., Polarization sensitive optical coherence tomography for materials characterization. *Chiang Mai Journal of Science*, 2018. 45: p. 2232-2237.
17. Chugh, T. and A.K. Jain, OCT Fingerprints: Resilience to Presentation Attacks. *ArXiv*, 2019. abs/1908.00102.
18. T. Chugh, A.K.J., OCT Fingerprints: Resilience to Presentation Attacks. *ArXiv*, 2019.
19. Akbari, N., Automation of Fingerprint Recognition Using OCT Fingerprint Images. *Journal of Signal and Information Processing*, 2012. 03: p. 117-121.
20. Moolla, Y., et al., Fingerprint Matching with Optical Coherence Tomography. 2015. 237-247.
21. Cimalla, P., et al., Simultaneous dual-band optical coherence tomography in the spectral domain for high resolution in vivo imaging. *Optics express*, 2009. 17: p. 19486-500.
22. Jain, A.K., Y. Chen, and M. Demirkus, Pores and ridges: high-resolution fingerprint matching using level 3 features. *IEEE Trans Pattern Anal Mach Intell*, 2007. 29(1): p. 15-27.
23. Perez, L. and J. Wang, The Effectiveness of Data Augmentation in Image Classification using Deep Learning. 2017.
24. Almahdi, R. and H. Ragb, Fused Deep Convolutional Neural Networks Based on Voting Approach for Efficient Object Classification. 2019.
25. Srivastava, N., et al., Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 2014. 15: p. 1929-1958.
26. Chollet, F., Xception: Deep Learning with Depthwise Separable Convolutions. 2017 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017: p. 1800-1807.
27. Szegedy, C., et al., Rethinking the Inception Architecture for Computer Vision. 2016 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016: p. 2818-2826.
28. Agarap, A.F., Deep Learning using Rectified Linear Units (ReLU). 2018.
29. He, K., et al. Deep residual learning for image recognition. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.
30. Simonyan, K. and A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR*, 2015. abs/1409.1556.
31. Li, Q., et al., Performance Evaluation of Deep Learning Classification Network for Image Features. *IEEE Access*, 2021. PP: p. 1-1.
32. Barredo Arrieta, A., et al., Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 2020. 58: p. 82-115.
33. Gilpin, L.H., et al. Explaining explanations: An overview of interpretability of machine learning. in *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*. 2018. IEEE.
34. Lundberg, S.M. and S.-I. Lee, A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 2017. 30.